Statistical Modelling for Quantitative Risk Analysis

Stolze, Sebastian

Award date:
2020

Awarding institution:
University of Bath

Statistical Modelling for Quantitative Risk Analysis

submitted by

Sebastian Stolze

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

September 2019

Declaration of any previous submission of the work

The material presented here for examination for the award of a higher degree by research has not been incorporated into a submission for another degree.

Signature of Author . . . . . .

Sebastian Stolze

Declaration of authorship

I am the author of this thesis, and the work described therein was carried out by myself personally.

Signature of Author . . . . . .

Sebastian Stolze

SUMMARY

Event trees are a common framework for quantitative risk analysis where a joint probability distribution is displayed in a tree format. They are closely related to Bayesian networks which utilise conditional independencies in their graph representation.

In this thesis we are mainly concerned with the connection between the two.

In the first part we propose a possible sequential translation algorithm of event trees to Bayesian networks. Tools from information theory are exploited to quantify the strength of dependencies within the network. This allows us to simplify the model by removing weak conditional dependencies. We apply this algorithm to smaller, artificial data and a real-world example. The algorithm can also be used to simplify a Bayesian network or to find the weakest link in it. One of the benefits of using a simpler Bayesian network is faster calculations.

In a next step we discuss two types of model extensions that can be included within the same type of algorithm. By construction, event trees can only model discrete variables and hence a derived Bayesian network will also show this limitation. Given our application background with safety risk, variables such as ignition time are represented as discretised versions in the event tree. We show how the ignition time variable can be made continuous in time and how possible direct consequence variables can be included given their dependence on a continuous set.

Lastly, we exemplify how loss data for an event tree can be used to make the translation / simplification context-dependent. Weighted entropy is used as a basis for a measure of similarity that accounts in a sense for both quantitative and qualitative differences. We contrast the different results using the entropy versus weighted entropy approach in the sequential algorithm on an artificial data set.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my academic supervisors, Evangelos Evangelou and Simon Shaw for their guidance, advice and especially their patience when things took a bit longer.

Further I would like to thank DNV GL for being the industrial partner in this project. In particular, thanks goes to David Worthington, my external supervisor, for taking the time to introduce me to the work DNV GL does and supporting me with answers to my questions and real-world data.

I am also grateful to Finn Lindgren for guiding me initially.

This project was partly funded by the Department of Mathematical Sciences at the University of Bath and partly by DNV GL.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

The initial problem setting for this project was born out of one of the regular workshops between academia and industry partners initiated by the Centre of Doctoral Training in Statistical Applied Mathematics in Bath. One of these industry partners is DNV GL, a global classification society with business areas such as amongst other things, oil and gas, maritime and digital solutions. DNV GL presented some of their interests related to event trees (ETs) within the context of quantitative risk analysis (QRA) and a group of students together with academic staff elicited possible project ideas. After initial analysis of typical weak points of ETs, the conclusion was drawn that the use of Bayesian networks (BNs) can provide an adequate model improvement.

This was the starting point for our work in which the research aim could be formulated as finding methods that improve the QRA capabilities of typical event tree models as used by DNV GL for applications within the oil and gas industry or related areas.

In a first stage of this project we examined ways an event tree can be reformulated into a Bayesian network. It is not uncommon that conditional independencies are found in ET data; they may not even be identified easily by manual inspection. In the literature e.g. (Marsh and Bearfield, 2008), a translation of ETs into BNs utilising direct checks of conditional independence is already described. However, it is also reasonable to examine the ET data for 'weak' conditional dependencies as a way of allowing further complexity reduction, hoping to not lose too much of the encoded distribution information. To quantify the information loss when treating a weak conditional dependence as a conditional independence, tools from information theory can be employed.

To address this idea, we present in Chapter 2 a sequential algorithm that allows to automatically translate and simplify an ET into a BN. Step-by-step the nodes are examined for the effect of a hypothetical removal of one or more of their parents by using conditional entropy and the conditional mutual information. We apply this algorithm to different ET data sets and even BNs in order to show its capabilities.

History shows that one of the key points in safety models is the ignition time; this is underlined

when looking at the translation of some of DNV GL's real-world ETs as we do in Chapter 2. Typically a number of conditions such as the type of material or the size of a leak influence the ignition time which in turn influences the different accident outcomes, such as whether there will be an explosion or not. Since ETs only include discrete variables (which might be discretised versions of continuous variables), we consider in Chapter 3 two continuous-time model extensions for the translated BNs that are related to ignition time. One of the extensions consist of re-modelling the ignition time variable using a piece-wise constant hazard function. The other extension concerned variables that are consequences of ignition time and so have a continuous-time underlying outcome set. We show how these extensions can be incorporated into the algorithm proposed in Chapter 2.

In safety risk analysis and other areas, it may be the case that the occurrence of events with small probabilities have a disproportionally large impact. This leads to the issue that whenever impact or loss data is available, probabilistic comparisons alone may be unsatisfactory. (Guiaşu, 1971) points out that "there exist many fields dealing with random events where it is necessary to take into account both these probabilities and some qualitative characteristics of events". In Chapter 4 we follow this thought and show briefly how the existing algorithm from Chapter 2 can be adapted to incorporate loss data by using weighted versions of information measures. We compare both approaches on one of the previous data sets.

In this first chapter we will set the stage for this work and introduce the basic objects and some of their properties.

1.1 Background on quantitative risk analysis

Quantitative risk analysis has been carried out for centuries. For example, some ancient cultures prepared for problems like droughts and floods by record keeping and taking preventive actions such as building storehouses, see (Covello and Mumpower, 1985) for a historical review of risk analysis. Nowadays, this could be compared to creating early versions of risk models, setting up guidelines and planning counter measures to limit negative impacts.

In the more recent history of QRA, around the 1960s and 1970s, probabilistic risk models became more fashionable. Early documented applications can be found in the areas of aerospace, nuclear power and chemical process industry. In the following paragraph, we briefly repeat and quote from (Bedford and Cooke, 2001) some points about the historical progression of probabilistic risk assessment in these areas.

One of the events sparking the development of more systematic methodologies in the aerospace industry includes the fire of the Apollo test AS-204 in 1967. Losses associated with this event were multi-dimensional. "This one event set the National Aeronautics and Space Administration (NASA) back 18 months, involved considerable loss of public support" and more importantly, cost three astronauts lives. Following up, efforts for quantitative risk methodologies at NASA "reached a high point with the publication of the SAIC Shuttle Risk Assessment" which demonstrated that

likelihoods of certain failures had been reduced, instead of eliminated (which may have been much harder or impossible). In the nuclear power industry, the need for quantitative methods grew in the 1950s as the framework of covering the 'maximum credible accident' and studying remaining risks by considering 'incredible accidents' proved unsatisfactory with regards to assessing such probabilities. It was desirable to quantify the improvements in the engineering process as changes in the impact of accidents. "The first full scale application of these methods [probabilistic risk assessment], including an extensive analysis of the accident consequences, was undertaken in the Reactor Safety Study WASH-1400 published by the US Nuclear Regulatory Commission (NRC). This study is rightly considered to be the first modern PRA [probabilistic risk analysis]." The study called some controversy, a number of experts disbelieving in the model numerics. The following decades lead to further improvements of the PRA framework with the aim to define tolerable risks.

There are different ways of assessing risks probabilistically, but a common general approach is to equate

$$\text{risk} = \text{likelihood} \times \text{impact}, \tag{1.1}$$

where likelihood is usually considered to be a set of probability distributions and impact could be any measure of harm deemed appropriate for the specific application. To work with such a framework it it necessary to estimate these probability distributions from data or to assign them through expert elicitation. (Fenton and Neil, 2013) name some common drawbacks related to (1.1) which include i) probabilities (especially small ones) are hard to estimate in practice and ii) impact can be measured on many scales, so that risks could at best be ordered according to their severity. These drawbacks become particularly relevant for industries where low probability - high impact accidents are a main concern, such as in aerospace, oil and gas and nuclear power.

It is not uncommon that the probability distributions considered for analysis are summarised using means or other functions to simplify inference or decision making processes. The process of summarising can be stretched too far or used in the wrong way, for example, by using unsuitable summary measures; this can lead to reasoning with sometimes catastrophic outcomes. One can also find in (Fenton and Neil, 2013) a number of small examples that demonstrate this problem.

There are types of models that consider more facets of the probability distributions which can circumvent over-simplification. Event trees are one class of such models. Their use dates back at least to the WASH-1400 nuclear power plant safety study in (Nuclear Regulatory Commission, 1975). They are commonly used to analyse hypothetical accidents by modelling consequences triggered by a certain initialising failure. There is a related concept called fault tree analysis (FTA) which often precedes event tree analysis (ETA). In fault tree analysis the aim is to systematically examine possible causes of failures. Together, FTA and ETA often form a so called bow-tie model, reminding of the shape the two tree models take. Further explanations can be found in e.g. (Vinnem, 2014). We will introduce ETs in more detail in the next section and meet some of their disadvantages.

1.2 Event trees as a type of quantitative risk model

1.2.1 The basic concept event tree

Event trees are a QRA tool that can be used to model, analyse and visualise a chain of events and their consequences. (Rausand, 2011) lists among other things that event tree analysis is aimed at: The identification of accident scenarios after a hazardous event, the determination of probability of each accident scenario, the determination and assessment of the consequences of each accident scenario. This is broadly the context in which we meet them in this work. Even though ETs have been around for a longer time, the first mathematical description seems to appear in (Papazoglou, 1998). We only give a short, intuitively accessible description of ETs here which is enough to understand the concept. A semantic model description of a 'complete' ET can be found in Appendix A.1. Hereby 'complete' ET shall mean an ET where every combination of outcomes, possible or impossible, is presented as a path in the tree.

ETs can be thought of as consisting of a combination of a structural component (a graph-theoretic tree) and a parametric component (conditional probability distributions).

To describe the structural component of ETs, we remind of some basic graph-theoretic concepts and present them as found in (Diestel, 2017) or (Wallis, 2007). We will require these concepts later on for Bayesian networks as well.

Definition 1.1. *(Graph; undirected, directed)*
An undirected graph G is a pair of sets $G = (V,E)$ with the following specifications. V is a non-empty set and $E \subseteq [V]^2 := \{\{x,y\}|x,y \in V, x \neq y\}$. That is, $[V]^2$ is the set of all unordered pairs made up from elements of V. The elements of V are called vertices (or nodes), the elements of E are called (undirected) edges. A directed graph G is a pair of sets $G = (V,E)$, where V is a non-empty set and $E \subseteq V \times V := \{(x,y)|x,y \in V, x \neq y\}$ is the set of (directed) edges.

In this thesis we assume that all graphs are directed graphs unless stated otherwise. The following definitions are then tailored to this special case of the underlying graph $G = (V,E)$ being assumed directed.

Definition 1.2. *(Path, connectivity, cycle, acyclic, distance)*
A (directed) path $< x_0, x_1, \ldots, x_{k-1}, x_k >$ between node $x_0 \in V$ and $x_k \in V$ is a sequence

$$(x_0, x_1, \ldots, x_{k-1}, x_k)$$

of $k+1$ distinct nodes ($k \geq 1$), such that two succeeding nodes x_i, x_{i+1} form an edge, i.e. $(x_i, x_{i+1}) \in E$. The vertices x_0 and x_k are said to be linked by the path, denoted by $x_0 \rightsquigarrow x_k$. We also say that $< x_0, x_1, \ldots, x_{k-1}, x_k >$ has length $| < x_0, x_1, \ldots, x_{k-1}, x_k > | = k$.

An undirected path $[x_0, x_1, \ldots, x_{k-1}, x_k]$ between node $x_0 \in V$ and $x_k \in V$ is a sequence

$$(x_0, x_1, \ldots, x_{k-1}, x_k)$$

of $k+1$ distinct nodes ($k \geq 1$), such that for every two succeeding nodes x_i, x_{i+1} either $(x_i, x_{i+1}) \in E$ or $(x_{i+1}, x_i) \in E$. The vertices x_0 and x_k are said to be connected by the undirected path, denoted by $x_0 \leftrightsquigarrow x_k$. From now on we shall assume the term path means directed path, unless stated otherwise.

The graph G is called connected, if any two of its vertices are connected by an undirected path in G.

A cycle $< x_0, x_1, \ldots, x_{k-1}, x_0 >$ is a sequence of distinct nodes $x_0, x_1, \ldots, x_{k-1}$, $k \geq 2$, of the type

$$(x_0, x_1, \ldots, x_{k-1}, x_0),$$

such that $(x_i, x_{i+1}) \in E$, for $i = 0, 1, \ldots, k-2$ and $(x_{k-1}, x_0) \in E$. (A cycle can be interpreted as a path where start and end are the same node). A graph that does not contain any cycles is said to be acyclic.

The distance $d_G(x, y)$ in G between the two nodes $x, y \in V$ that are linked by a path (connected by an undirected path) is the length of the shortest path linking x and y (the length of the shortest undirected path connecting x and y).

Definition 1.3. *(Parent, child, co-parent, descendant)*
We will occasionally denote a directed edge (x, y) by $x \to y$ and say that x is a parent of y and y is a child of x. These names will become clearer later. One can define the parent set $\mathrm{pa}(y)$ *of y to be the set* $\mathrm{pa}(y) := \{x | (x, y) \in E\} = \{x | x \to y\}$*, i.e. the set of all nodes x for which there exists a directed edge (x, y) in the graph. The set of children of x is then similarly defined as* $\mathrm{ch}(x) := \{y | (x, y) \in E\} = \{y | x \to y\}$.

Sometimes one talks about co-parents of x. This is simply the set of nodes sharing a child with x: $\mathrm{co}(x) := \{w | \exists y : w, x \in \mathrm{pa}(y)\}$.

A node $y \neq x$ is a descendant of a node x, if there exists a path that links x and y, i.e. $x \rightsquigarrow y$. If no such path exists, then y is a non-descendant of x. The set of descendants of x is $\mathrm{desc}(x) := \{y | y \neq x, x \rightsquigarrow y\}$ *and correspondingly, the set of non-descendants of x is* $\mathrm{ndesc}(x) := V \setminus \{\mathrm{desc}(x) \cup x\}$.

A popular special case of graphs are rooted trees. They exhibit a simple edge structure, such that every node (except for the root) only has one parent. From now on, whenever we write tree, it is understood to be a rooted tree.

Definition 1.4. *((Rooted) tree, generation)*
We define a rooted tree T to be a directed graph such that one arbitrary node $x_r \in V(T)$ of the graph is chosen to be called root and the following properties are fulfilled: i) all edges lead away from the root node, i.e. x_r is the only node such that there is no $x_i \in V$ with $(x_i, x_r) \in E$ and ii) for every other node x_i in T, there is one and only one path from x_r to x_i.

We say that the nodes $x \in V$ with $d_T(x_r, x) = k$ make up generation k of the tree:
$V_k := \{x \in V(T) : d_T(x_r, x) = k\}$. Furthermore, a tree node that has no children is called leaf node.

We can now define the concept event tree as follows.

Definition 1.5. *(Event tree)*
Let X_0 be a random variable with only one certain outcome in its outcome set $\mathscr{X}_0 := \{x_0\}$. We can imagine that X_0 describes the state of the world, that is, any certain background knowledge we have. Let $X_1,\ldots,X_n$ be discrete random variables where, for each X_i, $i = 1,\ldots,n$, the set of outcomes is

$$\mathscr{X}_i \quad = \quad \{x_{i1}, x_{i2}, \ldots, x_{im_i}\}.$$

We define the joint outcome space of $\{X_0, X_1, \ldots, X_n\}$ to be $\mathscr{X} := \mathscr{X}_0 \times \mathscr{X}_1 \times \ldots \times \mathscr{X}_n$. The graphical component of the event tree is a rooted tree structure that represents the joint outcome space $\mathscr{X}$ in the following way. The root represents X_0 and two additional conditions are fulfilled: i) every node of generation k represents an outcome in $\mathscr{X}_k$ and ii) every joint outcome $\boldsymbol{x} \in \mathscr{X}$ can be associated with one and only one path from the root to a node in generation n.

Suppose the joint mass function $p_{(X_0,X_1,\ldots,X_n)}$ on $\mathscr{X}$ is specified through a list of conditional probability functions

$$p_{X_0}(x_0) = 1, p_{(X_1|X_0)}, p_{(X_2|X_1,X_0)}, \ldots, p_{(X_n|X_{n-1},\ldots,X_1,X_0)}.$$

The parametric component of the event tree consists of this list of conditional probability mass functions, where the value $p(x_i|x_{i-1},\ldots,x_1,x_0)$ is attached to the edge (x_{i-1},x_i) of the path $(x_0,x_1,\ldots,x_{i-1},x_i)$.

Remark 1.6. *The ET description we have given is a construction for a 'full tree'. By 'full tree' we mean that every single $\mathbf{x} \in \mathscr{X}$ is represented by a path. This is often not the case in practice for two reasons: i) A subsequence $(x_0,x_1,\ldots,x_i)$ might render some $x_{i+1} \in \mathscr{X}_{i+1}$ impossible such that there is zero probability of observing it. Edges or parts of paths that represent such a situation need not be present. We keep them, but will assign probability zero to them. ii) Some outcomes might be judged irrelevant. If a branch of an ET leads to the exact same final consequence, then we do not need to care about the exact outcome at that branch. Example 1.10 shows how consequences associated with a path can be used in practice. (Marsh and Bearfield, 2008) also show some examples for what they call a 'don't care' condition.*

For ease of description, demonstration and implementation however, we decided to work with a 'full tree' construction. (Papazoglou, 1998) describes how a reduced representation can be obtained from the full joint outcome space.

By the above construction, we can see that an ET encodes the probability of a joint outcome $(x_0,x_1,\ldots,x_{n-1},x_n)$ as the product

$$p(x_0,x_1,\ldots,x_{n-1},x_n) = \prod_{i=1}^{n} p(x_i|x_{i-1},\ldots,x_1). \tag{1.2}$$

The event tree concept is intuitively very accessible and does not hide any technical complica-

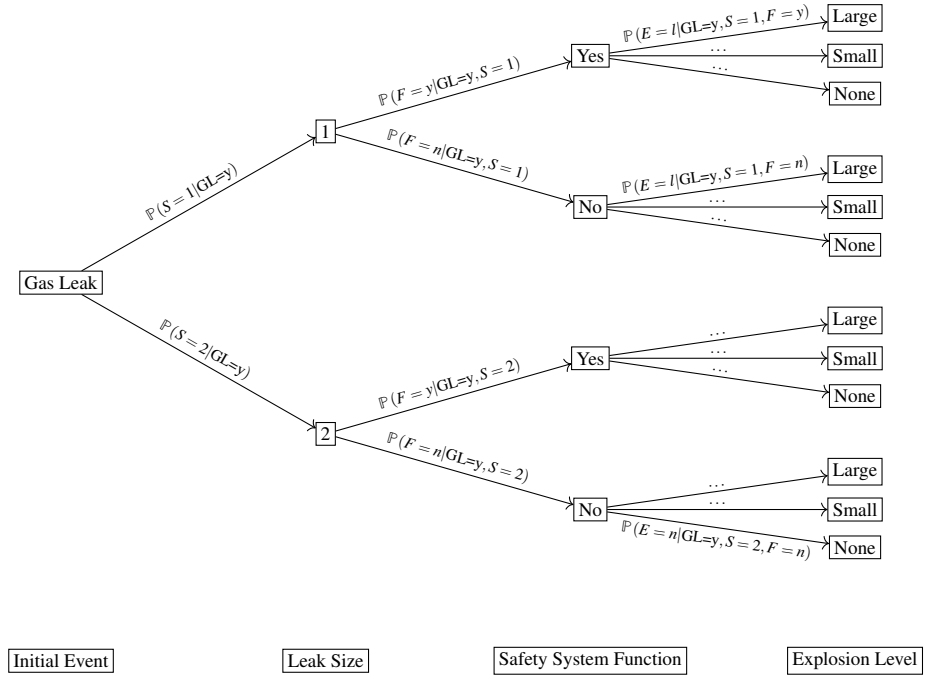

Figure 1-1: *Event tree for Example 1.7.*

tions. We show a small example motivated by our QRA theme.

Example 1.7. *Let us assume we model a possible gas leak scenario in a section of some oil platform, which may occur in one of two types, i.e. in type '1' or '2' . Dependent on that, our safety system might function or not, that is, the state of safety system function can be 'Yes' or 'No'. A direct consequence of the outcomes of the previous two variables may be the level of a possible explosion which could be 'large', 'small', 'none'. The corresponding event tree, together with the unspecified probabilities, would look like the one in Figure 1-1.*

We can now calculate the joint probability of a certain accident scenario, i.e. a sequence of outcomes. For example, we might ask for the probability of the following happening: given the gas leak has occurred, we have a leak of size 1 and a functioning safety system, resulting in a small explosion. This event in the joint outcome space will be denoted by $\{GL=y, S=1, F=y, E=s\}$.

Multiplication of the conditional probabilities along the path that describes exactly this scenario gives

$$\mathbb{P}(GL=y,S=1,F=y,E=s) = \\ \mathbb{P}(GL=y)\,\mathbb{P}(S=1|GL=y)\,\mathbb{P}(F=y|GL=y,S=1)\,\mathbb{P}(E=s|GL=y,S=1,F=y).$$

We should comment on two practical issues here.

Remark 1.8. *We note that the number of conditional probabilities to be specified in an ET grows rapidly with the number of variables modelled and the number of outcomes per variable. (This is*

even the case if edges with zero probability are omitted, unless the tree would be very sparse.) For a discrete probability distribution on a set of m many points, one needs to specify $m-1$ probabilities to fully determine the distribution. Hence, for a tree construction with all edges present, one needs to specify

$$(m_1-1)+(m_1(m_2-1))+\ldots+(m_1m_2\ldots m_{n-1}(m_n-1))=\sum_{i=1}^{n}(m_i-1)\prod_{j=1}^{i-1}m_j$$

many edge probabilities. In the above example this equals a total of eleven.

Remark 1.9. *In many applications, the variable X_0 is associated with a frequency that describes the occurrence of x_0 per year. In this case, multiplying this frequency with the probabilities along a specific path will give the frequency of occurrence of the joint outcome per year instead of the probability of the joint outcome.*

The literature on event trees and their applications is vast. We mention a few articles as representatives of the field.

Since its seemingly first methodical appearance in (Nuclear Regulatory Commission, 1975), ETA has been used in various areas. For example, (Hong et al., 2009) work with an application for an underwater tunnel excavation project. Their model has been used to "quantify the risk at the preliminary design stage of the tunnel" and to adapt safety countermeasures. (Neri et al., 2008) developed an ET to model risks related to eruption events of the volcano Vesuvius. They constructed the model through a structured expert elicitation procedure "to complement more traditional data analysis and interpretative approaches". (Rosqvist et al., 2013) use an ET in the context of flood protection in Finland where costs are the considered consequences. Aims of this project included to assess funding and prioritisation of flood protection measures.

We commonly find event trees as part of a bow-tie approach, such as in (Abimbola et al., 2014), where a somewhat dynamic version is applied to the area of offshore drilling. Here ET probabilities are updated using Bayes' theorem where observations are based on ongoing process measurements.

In the next subsection we briefly touch upon the topic of assessing risks using ETs.

1.2.2 Risk assessment using event trees

We outline how ETs are often used in safety risk as we believe that there are probably few different ways of using them in general.

For most of this thesis we are concerned with probabilistic structures of ETs and BNs and their approximation in certain ways. In practice it is common to use approaches similar to Equation 1.1 to quantify risk. Hence, besides probabilistic structures, there should be additionally considered an appropriately chosen quantity associated with impact or harm.

Every path from root to a leaf node in an ET can be thought of as an accident scenario or a chain of event outcomes associated with a hypothetical accident. For every such path we can

measure or hypothesise the resulting impact or harm. In safety risk this is usually the loss of life (LoL), i.e. how many people would have died in a certain development of an accident.

Let us consider a specific numeric example in order to illustrate the usage of such loss data.

Example 1.10. *Figure 1-2 displays a version of the ET from Example 1.7 where numerical probabilities have been assigned to the edges and an associated loss of life for each root-to-leaf path has been assigned.*

According to Equation 1.1 we might compute the risk associated with a specific scenario (a path from 'Initial Event' to 'Explosion level') by multiplying its likelihood (the joint probability as in Example 1.7) by the corresponding impact, in this case the given 'Loss of Life' data. In Example 1.7 we considered the event $\{GL = y, S = 1, F = y, E = s\}$ *as a scenario. Given the numerical probabilities we compute the likelihood as*

$$\mathbb{P}(GL = y, S = 1, F = y, E = s) = 0.8 \times 0.98 \times 0.02 = 0.1568.$$

Given a LoL for this scenario of 0.02*, one could compute*

$$risk(GL = y, S = 1, F = y, E = s) = 0.1568 \times 0.02 = 0.003136.$$

This does not mean much in itself, but similar calculations for all paths of the ET lead to a 'risk' for all scenarios which can be meaningfully summarised by stating the expected LoL. This is simply the sum of the 'risks' for each path, since this will be the sum of fatalities weighted by their probability. In this example the number of expected fatalities, given a gas leak would equal to $\mathbb{E}LoL = 0.1063528$.

Now this expected LoL is conditional on a gas leak happening. In practice, one works with frequencies of the initial event to express the expected losses per year. This frequency can be called accident frequency. We can imagine that the initial event (that other events are conditioned on) that we assigned probability one is given a frequency of occurrence per year. Then the joint probability of a path through the tree can be associated with a frequency per year when the joint probability is multiplied by the accident frequency. For the sake of this example, we can use a gas leak frequency of 10^{-4} *per year. Then we would estimate the frequency of the scenario* $\{GL = y, S = 1, F = y, E = s\}$ *to be* $10^{-4} \times 0.1568 = 1.568^{-5}$ *per year and the expected loss of life to be* $10^{-4} \times 0.1063528 = 1.063528 \times 10^{-5}$ *per year.*

To determine whether a risk structure is acceptable or not it is common to use so called $F-N$ *curves. (See for example (De Vasconcelos et al., 2015).) These curves display the relation between the number N of fatalities and the cumulative frequency* f_N *of N or more fatalities per year on a log-log scale. Usually there are ranges or bands for these curves that deem a risk structure acceptable.*

We give the $F-N$ *curve for our example in Figure 1-3, assuming again that the gas leak has a probability of occurrence of* 10^{-4} *per year. We can see for instance it is estimated that there are* 1.5 *or more fatalities approximately* 2.41×10^{-6} *times per year. Changes in the probabilities of*

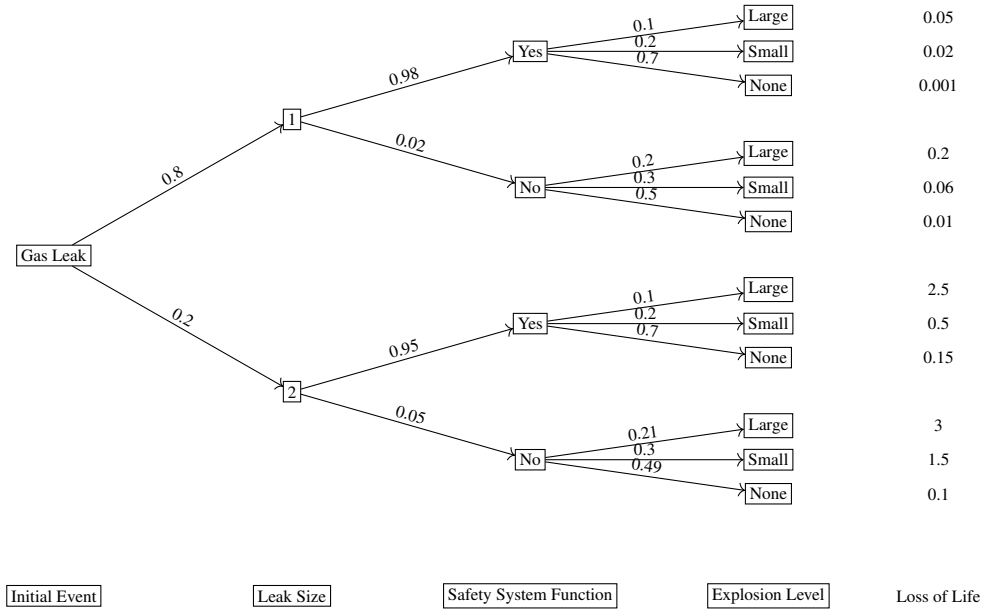

Figure 1-2: *Event tree for Example 1.10 including loss data.*

the ET and the LoL will affect these curves so that, for example, potential mitigation procedures can be compared quantitatively.

Let us now consider some of the problems that arise with the use of ETs.

1.2.3 Problems and an alternative risk model

Let us consider what some typical real-world ETs look like. We display an example supplied by DNV GL[1], a classification society that offers amongst other things QRA software to model risks for on- and offshore facilities and that supported this project. The company uses ETs to evaluate and analyse probabilities and results that were obtained from a complex, physical model. Part of a typical output from DNV GL's Safeti software[2] can be seen in Figure 1-4. A similar ET that was converted to table format and contains some additional information is shown in Figure 1-5.

Such a table usually consists of several thousand rows. Each row of the table corresponds to exactly one path through the ET. The columns of the table stand alternately for the nodes and edges, the node columns contain the outcomes of the random variables and the edge columns contain the corresponding conditional (edge) probabilities.

The typical ET files we have encountered contain variables such as the one above. For instance, `EventNo` contains a numbering of the different types of the hazardous initial event, `Weather` and `Wind` the weather state and the wind direction. We should note that usually also two types of frequencies occurred. `EventFreq` describes the occurrence of a type of initial event per year and `OutcomeFreq` describes the frequency per year of the type of initial event combined with a certain

[1] https://www.dnvgl.com/ - accessed 27 May 2019

[2] https://www.dnvgl.com/services/offshore-qra-safeti-offshore-1727 - accessed 27 May 2019

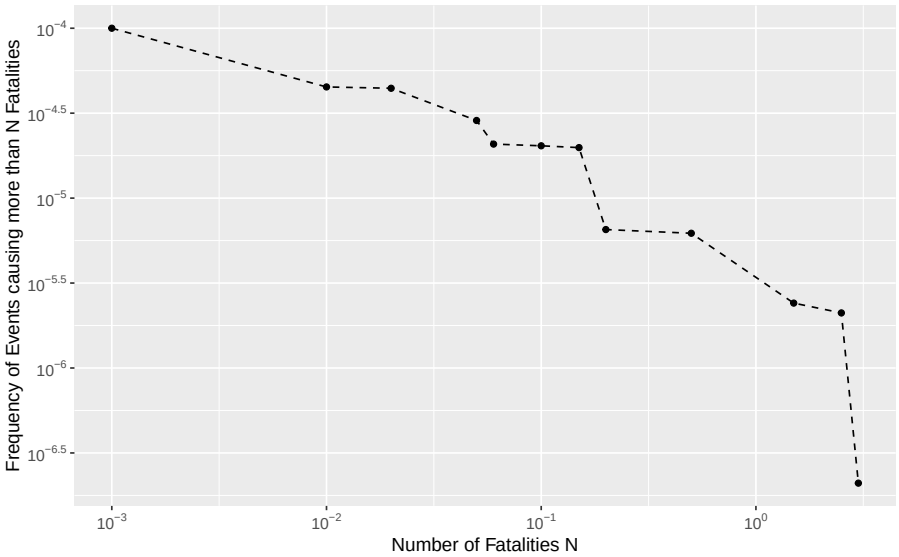

Figure 1-3: *F-N curve for Example 1.10.*

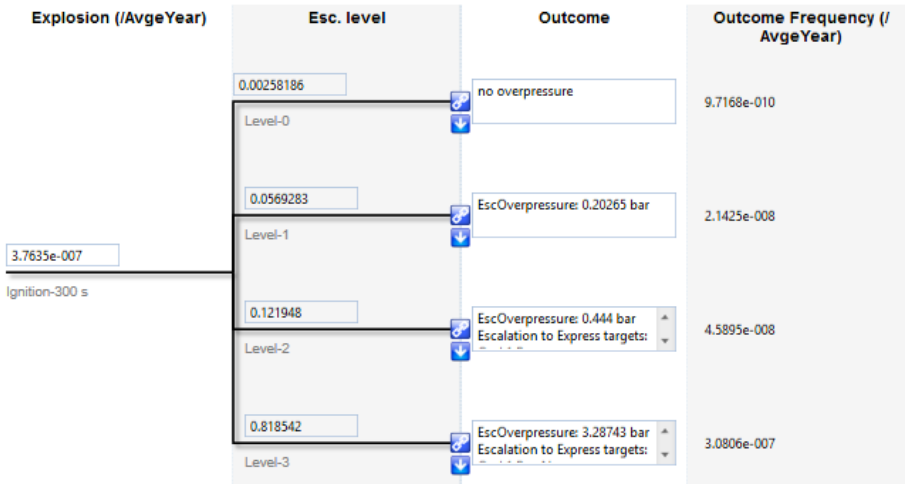

Figure 1-4: *Extract from a collapsed branch of a typical event tree as displayed in DNV GL's Safeti software.*

iWea	dWea	iDir	dDir	iTime	dTime	dProbImmDel	dProbNi	dProbi	iExp	dOPressin	dPExp	iFireWater	dPFireWater	iEarly	dPEarly	iHVAC	dPHVAC	sType	dETFreq
1	0.5205199	1	0.120824967	1	0	0	1	1.00E-02	1	0	1	TRUE	1	FALSE	0.02	1	0.2	CNdEFO	1.51E-20
1	0.5205199	1	0.120824967	1	0	0	1	1.00E-02	1	0	1	TRUE	1	FALSE	0.02	0	0.8	CNdEFO	6.06E-20
1	0.5205199	1	0.120824967	1	0	0	1	1.00E-02	1	0	1	TRUE	1	TRUE	0.98	1	0.2	CNdEFO	7.42E-19
1	0.5205199	1	0.120824967	1	0	0	1	1.00E-02	1	0	1	TRUE	1	TRUE	0.98	0	0.8	CNdEFO	2.97E-18
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	1	20000	0.829750379	TRUE	1	FALSE	0.02	1	0.2	CNdEXO	5.50E-21
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	1	20000	0.829750379	TRUE	1	FALSE	0.02	0	0.8	CNdEXO	2.20E-20
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	1	20000	0.829750379	TRUE	1	TRUE	0.98	1	0.2	CNdEXO	2.69E-19
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	1	20000	0.829750379	TRUE	1	TRUE	0.98	0	0.8	CNdEXO	1.08E-18
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	2	40000	0.075109362	FALSE	1	FALSE	0.02	1	0.2	CNdEXO	4.98E-22
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	2	40000	0.075109362	FALSE	1	FALSE	0.02	0	0.8	CNdEXO	1.99E-21
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	2	40000	0.075109362	FALSE	1	TRUE	0.98	1	0.2	CNdEXO	2.44E-20
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	2	40000	0.075109362	FALSE	1	TRUE	0.98	0	0.8	CNdEXO	9.75E-20
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	3	50000	0.084333118	FALSE	1	FALSE	0.02	1	0.2	CNdEXO	5.59E-22
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	3	50000	0.084333118	FALSE	1	FALSE	0.02	0	0.8	CNdEXO	2.23E-21
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	3	50000	0.084333118	FALSE	1	TRUE	0.98	1	0.2	CNdEXO	2.74E-20
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	3	50000	0.084333118	FALSE	1	TRUE	0.98	0	0.8	CNdEXO	1.10E-19
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	4	78552.17432	8.07E-04	FALSE	1	FALSE	0.02	1	0.2	CNdEXO	5.35E-24
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	4	78552.17432	8.07E-04	FALSE	1	FALSE	0.02	0	0.8	CNdEXO	2.14E-23
1	0.5205199	1	0.120824967	2	30	0	0.99	4.42E-03	4	78552.17432	8.07E-04	FALSE	1	TRUE	0.98	1	0.2	CNdEXO	2.62E-22

Figure 1-5: *Snippet (not containing all columns) from an event tree output table of DNV GL's Safeti software.*

path through the ET, that is, a certain scenario. It is obtained as the row product of `EventFreq` and all the conditional probabilities across one row. `EventFreq` was converted to a probability distribution by normalising every frequency by the sum of the frequencies of all different initial events; then `OutcomeFreq` becomes a probability distribution as well, by construction.

In the last subsection we showed that the number of branches / outcomes to be displayed grows multiplicatively with the number of variables and sizes of the outcome sets. That means event trees (such as trees in general) grow very quickly in size and become hard to visualise. This issue translates into very large table files that often contain redundant information. It is hard to grasp possible (conditional) independencies within the tree and to organise this information in a simple manner. We identify some other drawbacks as well.

Once an event tree is constructed, it is a rather static object in the following sense. Suppose we want to introduce another random variable (perhaps somewhere in between the existing chain of consequences) to our model. Then we had to adjust a larger part of the already existing tree structure and change corresponding conditional probabilities.

In Figure 1-5 we see a column named `dTime`, which is a discretised time-to-event variable. (In our type of data sets, this is a time to ignition.) Event trees are discrete mathematical objects and hence cannot handle continuous type objects explicitly.

By using ETs, risk is commonly summarised as an expected loss by multiplying probabilities for final outcomes by some impact or utility. A model where the distribution of risks, or at least, different conditional distributions can be displayed explicitly and compactly would be preferable.

We suggest Bayesian networks (also called belief networks) as a viable alternative to ETs because of their close relationship and because they remedy these issues and have some further advantages. For example, (i) a network representation allows to display dependencies and complexities of the joint distribution more compactly, (ii) 'modular storage' of the joint distribution should allow for simpler inference calculations in different directions or (iii) availability of software packages.

The use of Bayesian networks for risk analysis and similar problems is an area of growing interest and with a wide range of applications, such as in credit and default modelling, genetic

models, sensor validation; see (Pourret et al., 2008) for a number of these examples. Other areas also include medical decision support (Sesen et al., 2013) or even sports betting (Constantinou et al., 2013).

The next section deals with Bayesian networks in more detail.

1.3 Bayesian networks

1.3.1 Introduction to Bayesian networks

Bayesian networks are a framework to (graphically) represent joint distributions by exploiting conditional independencies. They consist of two parts - i) a qualitative part, given by a directed acyclic graph (DAG) that displays some conditional independence relations among the random variables that are represented by its finite node set $\{X_1,\ldots,X_n\}$ and ii) a quantitative part given by a number of conditional probability distributions (CPDs). Often enough, when all the variables are discrete, we are given conditional probability tables (CPTs).

The first systematic development of the theory of BNs is to our knowledge by (Pearl, 1988). We exhibit a mathematical definition of a Bayesian network, adapted from (Koski and Noble, 2009).

Definition 1.11. *(Bayesian network)*
A Bayesian network is a pair (G,p), where $G=(V,E)$ is a directed acyclic graph with node set $V=\{X_1,X_2,\ldots,X_d\}$, for some $d\in\mathbb{N}$, E is the edge set and p is either a probability distribution or a family of probability distributions, indexed by a parameter set Θ, over d random variables $\{X_1,X_2,\ldots,X_d\}$. The pair (G,p) satisfies the following criteria:

1. *For each $\theta\in\Theta$, $p(\cdot|\theta)$ is a probability function over the same state space $\mathscr{X}$. That is, for each $\theta\in\Theta$, $p(\cdot|\theta):\mathscr{X}\to[0,1]$ and $\int_{\mathscr{X}} dp(\mathbf{x}|\theta)=1$.*

2. *For each node $X_v\in V$ with no parent variables, there is assigned a probability distribution p_{X_v} of the random variable X_v. To each node $X_v\in V$ with a non-empty parent set $\mathrm{pa}(X_v)=\left\{X_{b_1^{(v)}},X_{b_2^{(v)}},\ldots,X_{b_m^{(v)}}\right\}$, there is assigned a conditional probability function $p_{X_v|\mathrm{pa}(X_v)}$ of X_v given the variables $\left\{X_{b_1^{(v)}},X_{b_2^{(v)}},\ldots,X_{b_m^{(v)}}\right\}$. If X_v has no parents, set $\mathrm{pa}(X_v)=\{\}$, the empty set, so that $p_{X_v}=p_{X_v|\mathrm{pa}(X_v)}$. The joint probability function p may be factorised using the (conditional) distributions $p_{X_v|\mathrm{pa}(X_v)}$ thus defined:*

$$p_{X_1,X_2,\ldots,X_d}=\prod_{v=1}^{d}p_{X_v|\mathrm{pa}(X_v)}. \tag{1.3}$$

3. *The factorisation is minimal in the sense that for an ordering of the variables such that $\mathrm{pa}(X_j)\subseteq\left\{X_1,X_2,\ldots,X_{j-1}\right\}$, $\mathrm{pa}(X_j)$ is the smallest set of variables such that*

$$X_j\perp\mathrm{pa}(X_j)^c\,|\mathrm{pa}(X_j).$$

That is, $\text{pa}(X_j) = \bigcap \left\{ A \subseteq \left\{X_1, X_2, \ldots, X_{j-1}\right\} : X_j \perp A^c | A \right\}$.

Remark 1.12. *We will use the terms 'variable' and 'node' interchangeably when suitable.*

For every node X_i, there is given a conditional probability function $p_{X_i|\text{pa}(X_i)}$ that describes the conditional probability of X_i attaining some value, given we know the instantiation of its parent nodes (random variables).

Note that any joint probability function $p_{X_1,X_2,\ldots,X_d}$ can be generally factorised just as in the case of an ET factorisation in (1.2):

$$p_{X_1,X_2,\ldots,X_d} = \prod_{i=1}^{d} p_{X_i|\{X_{i-1},X_{i-2},\ldots,X_1\}}. \tag{1.4}$$

If we compare (1.4) to the BN factorisation (1.3), we see that the BN factorisation uses a potentially smaller set of conditioning variables $\text{pa}(X_i)$ for each factor.

Given that our starting point are ETs, let all variables be discrete for the moment. Let us assume that there is an ordering of the nodes such that $\text{pa}(X_i) \subseteq \{X_1, X_2, \ldots, X_{i-1}\}$ for all $i = 1, \ldots, d$. Consider a fixed $i \in \{1, \ldots, d\}$ and $j \in \{1, \ldots, i-1\}$. Suppose that for every possible $(x_1, x_2, \ldots, x_j, \ldots, x_i) = \mathbf{x}_{1:i} \in \mathscr{X}_1 \times \ldots \times \mathscr{X}_i$ and for every $\tilde{x}_j \in \mathscr{X}_j$ it holds

$$p_{X_i|\{X_{i-1},X_{i-2},\ldots,X_1\}}(x_i|x_{i-1},\ldots,x_j,\ldots,x_1) = p_{X_i|\{X_{i-1},X_{i-2},\ldots,X_1\}}(x_i|x_{i-1},\ldots,\tilde{x}_j,\ldots,x_1),$$

then there is a conditional independence of the type

$$X_i \perp X_j | \{X_{i-1}, \ldots, X_1\} \setminus X_j$$

encoded. In the definition of a BN, this conditional independence was present when the DAG had no edge of the type (X_j, X_i). Thus, the structure of missing arcs in the DAG encodes conditional independence statements of the joint distribution. On the other hand, not all conditional independence statements might be directly seen by the DAG.

Before looking at an example, we shortly introduce the following notions and Markov-type properties that emphasise how a BN structure benefits our understanding of the joint distribution it represents.

One regularly appearing concept in the area of BNs is d-separation. The following definition is an adaption from (Scutari and Denis, 2015).

Definition 1.13. *(d-separation)*
If X, Y and Z are three disjoint subsets of nodes in a DAG G, then Z is said to d-separate X from Y, denoted $X \perp_d Y | Z$, if along every undirected path between a node in X and a node in Y there is a node v satisfying one of the following conditions:

(i) $v \in Z$ and does not have converging arcs, i.e. there are no consecutive nodes u, v, w as part of the undirected path, such that $(u, v) \in E$ and $(w, v) \in E$;

(ii) $v \notin Z$, every descendant of v is not in Z and v has converging arcs.

d-separation is a way of establishing conditional independencies in BNs. For instance, the following is proven in (Koski and Noble, 2009) for discrete distributions.

Theorem 1.14. *Let G be a DAG and let p be a probability distribution that factorises along G. Then for any three disjoint subsets $X,Y,Z \subset V(G)$, it holds that $X \perp_d Y|Z \Rightarrow X \perp Y|Z$.*

This property is also known as the Global Markov property. One can find an overview of different Markov-type properties for graphical models in (Forré and Mooij, 2017). We only state the following two.

Definition 1.15. *((Directed) Global Markov property)*
Let G be a DAG with $V = \{X_1,\ldots,X_n\}$ and p a probability distribution on $\mathscr{X} = \mathscr{X}_1 \times \ldots \times \mathscr{X}_n$. p is said to fulfil the global Markov property with respect to G, if for all disjoint subsets $X,Y,Z \subset V$:

$$X \perp_d Y|Z \Rightarrow X \perp Y|Z.$$

Definition 1.16. *((Directed) Local Markov property)*
Let G be a DAG with $V = \{X_1,\ldots,X_n\}$ and p a probability distribution on $\mathscr{X} = \mathscr{X}_1 \times \ldots \times \mathscr{X}_n$. p is said to fulfil the local Markov property with respect to G if for every $X \in V$:

$$X \perp \mathrm{ndesc}(X)|\mathrm{pa}(X).$$

For BNs, both properties are equivalent as was shown by (Lauritzen et al., 1990).

Theorem 1.17. *(Equivalence of Markov properties for BNs)*
For any DAG G, the global Markov property is equivalent to the local Markov property.

A related concept that frequently appears in the area of BNs is that of a Markov blanket.

Definition 1.18. *(Markov blanket)*
The Markov blanket $\mathrm{mb}(X)$ of a node X is given by the set of parents, children and co-parents of X:

$$\mathrm{mb}(X) := \mathrm{pa}(X) \cup \mathrm{ch}(X) \cup \mathrm{co}(X).$$

The Markov blanket of a variable X d-separates it from all other variables in a BN, hence it can be used to establish conditional independence. Given the information from $\mathrm{mb}(X)$, X is conditionally independent of all other variables: $X \perp \{Y \in V(G) : Y \notin \{X\} \cup \mathrm{mb}(X)\}|\mathrm{mb}(X)$.

Lastly, we comment that different factorisations of a distribution can encode the same conditional independence statements. A simple example demonstrates this idea.

Example 1.19. *(Different DAGs, but same conditional independence statements)*
This is a well known example and can be found for example in (Koski and Noble, 2009). Consider a distribution p_{X_1,X_2,X_3} over $\{X_1,X_2,X_3\}$ with the factorisation $p_{X_1,X_2,X_3} = p_{X_1}p_{X_2|X_1}p_{X_3|X_2}$ over the left DAG and factorisation $p_{X_1,X_2,X_3} = p_{X_2}p_{X_1|X_2}p_{X_3|X_2}$ over the right DAG.

Both DAGs imply the same conditional independence statements $X_1 \perp X_3 | X_2$ and $X_3 \perp X_1 | X_2$. (The two statements are actually equivalent due to the symmetry property of conditional independence.) Thus two DAGs with different structure may imply the same set of conditional independences.

Let us illustrate a Bayesian network with a small example from (Korb and Nicholson, 2011). It states a simplified version of the 'Asia' problem as appeared in (Lauritzen and Spiegelhalter, 1988). In the original example from (Lauritzen and Spiegelhalter, 1988), a patient that visited Asia is presented to a doctor with dyspnoea and the doctor would like to assess the chances of lung cancer or tuberculosis, given some patient background and test results. The authors remark that this model is stylised, but illustrative for the nature of a BN.

Example 1.20. *(Modified Asia network)*
In this example there are five variables each of which can be in two states. The variables are called

$$\texttt{Pollution, Smoker, Cancer, Dyspnoea, X-ray}$$

and the state values are called $\{\text{high}, \text{low}\}$ *and* $\{\text{true}, \text{false}\}$ *respectively.*

The structure of the BN comes from what we believe are causal relations. The CPTs for each node could have been elicitated from an expert in the field and / or historical data. See Figure 1-6 for the network structure and Figure 1-7 for the CPTs. This model tries to investigate questions of the type 'What is the chance a patient has cancer, given we have information about, say, her smoking habits and her X-rays'.

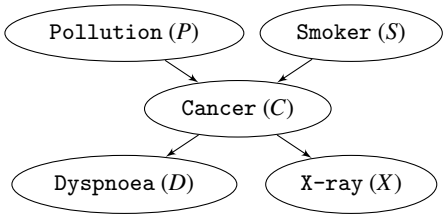

Figure 1-6: *Structure of the 'Asia' BN in Example 1.20.*

$\mathbb{P}(P=L)$	$\mathbb{P}(S=T)$
0.9	0.3

P	S	$\mathbb{P}(C=T\|P,S)$
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

C	$\mathbb{P}(X=T\|C)$
T	0.9
F	0.2

C	$\mathbb{P}(D=T\|C)$
T	0.65
F	0.3

Figure 1-7: *CPTs for the 'Asia' BN in Example 1.20.*

The Markov blanket for the variable `Dyspnoea` *consists only of its parent variable* `Cancer`*, so that given* `Cancer`*,* `Dyspnoea` *is conditionally independent of every other variable.*

Once a BN has been defined, it can be used to carry out different kinds of inference. Consider Example 1.20 and suppose we have information about the leaf nodes, that is, observation about test results and we ask for the probability of a patient having cancer. This kind of inference can be considered 'backwards' or explanatory. If, on the other hand, we only have information about the patients background and we ask for his chance of having cancer, the direction of the inference is 'forward' or predictive.

More generally, given a BN, we can supply observations about a subset of variables $\mathbf{X_O} \subset V(G)$ by setting $\mathbf{X_O} = \mathbf{x_O}$ and then ask for updated posterior distribution of a second subset of variables $\mathbf{X_Q} \subset V(G)$ given information about $\mathbf{X_O}$. This query for $p(\mathbf{X_Q}|\mathbf{X_O} = \mathbf{x_O})$ can be called the inference problem.

1.3.2 Short overview of the literature

The academic literature on Bayesian networks is vast. A search for the term 'Bayesian network' on Google scholar gives roughly 1.69 million results, as of Spring 2019.

Most literature focusses on the cases of networks with only discrete random variables (discrete BNs) or networks with both discrete and continuous variables (hybrid BNs) that have a special structure. Non-parametric versions of BNs have also been presented (Hanea and Kurowicka, 2008); here a network is set up by specifying rank correlations instead of conditional probabilities and inference is carried out by simulation. The structural restriction for most hybrid BNs seems to contain the conditions that i) discrete nodes do not have continuous parent nodes and ii) the continuous variables being (conditionally linearly) normally distributed. This usually allows for simpler, exact computational procedures. The literature can broadly be classified as dealing with either inference aspects, model learning, model extensions or applications of BNs.

Inference algorithms can be divided into exact inference and approximate inference. Exact inference algorithms are available mostly only to the above two cases (discrete or special hybrid networks). Most prominent for discrete networks seems to be the algorithm from (Lauritzen and Spiegelhalter, 1988) which relies on junction trees where 'messages' are passed between cliques of an associated graph. (A clique is a subset $C \subseteq V(G)$ of vertices of an undirected graph G, such that for all pairs of distinct nodes $x, y \in C$ there is an edge between x and y that comes from $E(G)$:

$\forall x, y \in C, x \neq y : \exists \{x, y\} \in E(G)$.) This algorithm is also used in the `R`-package `gRain` (Højsgaard, 2012) which we will use later on. Approximate inference algorithms are often based on one of the following: distributional approximations where a complicated distribution is approximated by a representative of a class of chosen and more tractable distributions and direct inference is carried out on the approximation; variational methods where an approximation of a distribution is usually found by iteratively solving some optimisation problem by varying candidate approximations from a class of distributions and evaluating a measure of divergence; or simulation-based methods using algorithms that will realise samples approximating a more complicated distribution. Distributional approximations can be made using for example mixtures of polynomials (Shenoy and West, 2011) or mixtures of truncated exponentials (Moral et al., 2001). For an introduction to variational methods for graphical models, see (Jordan et al., 1999). We can find an example of simulation based inference using Gibbs-sampling in (Hrycej, 1990). A review of different inference methods for BNs can be found in (Salmerón et al., 2018).

Considerable interest has been developed in learning both the structure as well as the parameterisations of BNs, given observational data. Structure learning largely includes algorithms based on conditional independence tests, e.g. the grow-shrink algorithm in (Margaritis, 2003) or scoring functions such as the K2-algorithm (Cooper and Herskovits, 1992). Learning the parameterisation of a BN requires a DAG to be given; one can use Bayesian methods to update parameter estimates, often using Dirichlet priors (Heckerman, 2008). There is a variety of learnings algorithms implemented in different `R`-packages, such as in `bnlearn` (Scutari, 2010).

Model extensions may contain various tweaks or generalisations of special cases. Dynamic versions of BNs exist mainly as Dynamic Bayesian Networks (DBNs), see for example (Mihajlovic and Petkovic, 2001), but also as continuous-time Bayesian networks (CTBNs) (Nodelman et al., 2002). DBNs can be thought of as chain of similar (sub-)networks where each subnetwork represents a time slice at one time point. These time-sliced subnetworks are linked by intertemporal links to form a chain. CTBNs can be thought of as networks where the state of each variable relates to a continuous-time Markov chain with a dependence on the states of its parent variables; the states of which in turn also depend on other continuous-time Markov chains.

Lastly, we want to mention some famous subclasses of BNs that have been applied to numerous problems. A simple, yet often powerful classification vehicle is the so called Naive Bayes Classifier (NBC) and its extensions, such as Tree Augmented Naive Bayes (TAN), see (Friedman et al., 1997). Even though these models rely on strong conditional independence assumptions, they often perform reasonably well in practice. Another case are Hidden Markov Models (HMM) which are used to model hidden states of a time-series like object and observations based on hidden states, see for example (Rabiner, 1989).

Given the interest in BNs, some problems in their usage have surfaced.

1.3.3 Disadvantages using Bayesian networks

There are a few difficulties related to Bayesian networks. We state some of the issues covered in the literature.

Inference in BNs can prove difficult. It is important to mention that (Cooper, 1990) showed that exact probabilistic inference for general BNs is NP-hard and (Dagum and Luby, 1993) prove that "the general problem of approximating probabilistic inference with belief networks is intractable in the worst case". Both these results provide reasonable justification to search for easily tractable special cases and to examine the effects of structural simplification methods for BNs.

For certain cases BNs have representational disadvantages. (Barclay et al., 2013) and (Barclay et al., 2015) refer to the fact that BNs cannot represent context-specific conditional independencies directly and that Chain Event Graphs (CEGs) might be a more sensible model if "only certain combinations of variables affect another variable and this cannot be represented simply by the directed edges between variables in the BN".

Lastly, we want to repeat the point made with Example 1.19. It is possible to have different DAGs along which the distribution is factorised that represent the same conditional independence statements. In that sense the DAG representation may not be unique.

1.4 Contributions of this work

The contributions of this project are threefold.

Firstly, we generalise the translation of event trees to Bayesian networks. The available literature considers a direct translation which includes checks for conditional independencies. We extend this method by introducing checks for weak conditional dependencies using an information measure that allows us to quantify the strength of these dependencies. This opens up the possibility to remove statistically weak connections to simplify the resulting structure. We describe a possible way of deciding how to set an acceptable loss for local approximations. Especially for distributions determined from simulations, where some numbers might be different because of numerical inaccuracies or precision, this could provide an automatic way of finding 'real' independencies. The suggested algorithm also allows to consider different information measures, dependent on the application background. We apply the algorithm for a specific case to a small scale, artificial ET, as well as to a modified real-world data set. The same method can be used to attempt to reduce the parent sets of any available Bayesian network. We also test a specific version of the algorithm on BNs from the literature.

Secondly, we examine some possible extensions in the ET / BN modelling and translation framework. The specific extensions are motivated by real-world QRA within the safety industry and consist of adding variables with a special structure, more precisely: i) The addition of a continuous-time node that is governed by a piece-wise constant hazard function and usually represents a time to ignition within the QRA applications. ii) The addition of variables with a finite number of states that have such a time-to-event variable from i) as a potential parent and have a

probability function of simple polynomial form. These variables usually represent binary consequences within the QRA applications. In a majority of the literature, the variables of a Bayesian network are discrete or of a simple Gaussian form. We add to the specific cases when these assumptions may not be the best choices. The necessary changes for a translation algorithm that works with these variables are presented.

Lastly, we address the choice of information measure and its implications for applications. The change of information measure allows to consider different ways of choosing a model approximation. Often one can only measure informational dissimilarity of models by comparing probability distributions. For the case where it is aimed at comparing the risk structure of two models and impacts / utilities are given, it may not be satisfactory to only compare probabilistic structures and then one should make full use of all information by employing a measure that utilises these impacts. We add to applications of the weighted entropy by another modification of the translation that reflects these thoughts and contrast between comparing probabilistic structures only and comparing probabilistic and impact structures.

CHAPTER 2

AN ALGORITHM FOR TRANSLATING AND SIMPLIFYING EVENT TREES TO BAYESIAN NETWORKS

The focus of this chapter lies on discrete random variables taking values in a finite subset of $\mathbb{R}$. For simplicity we will state any definitions or theorems for this special case, but remark whenever generalisations are available and needed for our purposes. A random variable X of this type will be taking values in $\mathscr{X}$ and have mass function p_X. For the specific probability of X being equal to $x \in \mathscr{X}$, we will either write $p_X(x)$ or $p(X = x)$ or even just $p(x)$, if the context is clear.

Event trees (ETs) and Bayesian networks (BNs) are both frameworks to represent a joint probability distribution over a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. We established previously that ETs factorise a joint probability of the type $p(x_1, x_2, \ldots, x_n)$ as

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{i-1}, x_{i-2}, \ldots, x_1), \tag{2.1}$$

where all factors of the product in (2.1) are given from the conditional probabilities along a specific path from the root of the tree to one of its leaves. This path is described by the tuple $(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$. On the other hand, a BN factorises this joint probability in the form

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | \mathrm{pa}(x_i)), \tag{2.2}$$

where $\mathrm{pa}(x_i)$ denotes the suitable instantiation of the parent set $\mathrm{pa}(X_i)$ of node X_i. (Informally, the 'relevant subset' of conditioned on states from $x_{i-1}, x_{i-2}, \ldots, x_1$.)

Remark 2.1. *The factorisation* (2.1) *holds in general and does not depend on the order of its variables. However, we can take the ordering implied by the generations of the ET as a natural order. For the construction of a BN there are usually different DAGs that encode the same conditional independence statements and so the factorisation* (2.2) *depends on an ordering* σ *of the variable indices* $\{1, 2, \ldots, n\}$*, that is, a bijection* $\sigma : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$*. Different orderings* $\sigma_{(1)}$*,*

$\sigma_{(2)}$ result in different factorisations using different conditional probabilities. We assume throughout the rest of this text that one ordering σ is fixed and whenever we want to create a BN from a given ET, then this ordering is given by the order of the generations of the relevant tree.

The following theorem due to (Shafer, 1996) is presented with proof in (Koski and Noble, 2009) and allows us to assume that $\mathrm{pa}(X_i) \subseteq \{X_1, X_2, \ldots, X_{i-1}\}$ from now on.

Theorem 2.2. *For any DAG with a finite number of nodes $X_1, \ldots, X_n$ there is an ordering $\left(X_{\sigma(1)}, \ldots, X_{\sigma(n)}\right)$ of the nodes (not necessarily unique) such that the parents of $X_{\sigma(i)}$ are a subset of $\{X_{\sigma(1)}, \ldots, X_{\sigma(i-1)}\}$. That is, by renaming the nodes as $W_j = X_{\sigma(j)}, j = 1, \ldots, n$, the parents of W_j are a subset of $\{W_1, \ldots, W_{j-1}\}$ for each $j = 1, \ldots, n$.*

Then, as (Koski and Noble, 2009) put it "For any joint probability distribution over a set of variables, with a given ordering for the variables, there is a directed acyclic graph over which the probability distribution may be factorized, where for each node $X_j, \mathrm{pa}(X_j) \subseteq \{X_1, \ldots, X_{j-1}\}$".

Equation 2.1 can be regarded as a special case of (2.2) where $\mathrm{pa}(X_i) = \{X_1, \ldots, X_{i-1}\}$. If the factorisation (2.2) encodes conditional independencies amongst its variables, i.e. for at least one $i \in \{1, \ldots, n\}$ it holds

$$\mathrm{pa}(X_i) \subsetneq \{X_1, \ldots, X_{i-1}\},$$

then this product becomes simpler than (2.1). Therefore, we will describe and demonstrate a sequential method that tests for encoded conditional independencies and additionally quantifies strength of dependencies using an information measure. Dependencies may be weak enough to ignore them for model simplicity. We try to simplify the factors in products like (2.1) and (2.2) by possibly removing conditioning variables with the aim to simplify these products as much as possible without losing too much information.

In the first section of this chapter we show how one can set up a Bayesian network corresponding to an event tree in a generic way without carrying out any conditional independence checks. Then we introduce some notions from information theory which are necessary to construct an algorithm that extends this translation by testing for conditional independencies (and quantifying 'weak' conditional dependencies) to create a Bayesian network that can be simplified according to a chosen threshold. This is an automated extension of the more common direct translations that usually only test for conditional independencies. We test the algorithm on both small, synthetic and larger real-world data sets given by event tree tables. In the end, we employ the algorithms simplification abilities for some Bayesian networks found in the literature.

2.1 Generic formation of Bayesian networks from event trees

We state with Algorithm 1 the simplest possible way of creating a BN from a given ET in a sequential manner. The resulting BN is maximal in the sense that the parent set of every node consists of all previous nodes.

Algorithm 1 Creation of a generic BN from a given ET.

Input: Event tree with root X_0 and n generations representing variables $X_1, X_2, \ldots, X_n$, with ordering σ given from the sequence of the generations and all conditional probabilities along the edges.
Output: Corresponding BN.

Let $V := \{\}$ and $E := \{\}$.
For $i = 1, \ldots, n$ do:

1. Create a node X_i and update the node set $V \leftarrow V \cup \{X_i\}$.
2. Define the parent set $\text{pa}(X_i) = \{X_1, \ldots, X_{i-1}\}$ and update the edge set
$$E \leftarrow E \cup \bigcup_{j=1}^{i-1} \{(X_j, X_i)\}.$$
3. Define a CPT for $p(X_i | X_{i-1}, \ldots, X_1)$ by using all edge probabilities of generation i of the tree.

Remark 2.3. *It should be remarked that we did not include the variable X_0 in the network. This variable can be seen as 'the state of the world' and all other variables considered can be seen as implicitly dependent on it.*

Since no simplifications are carried out and every node has all previous variables (according to σ) as parent set, this represents the case where factorisations (2.1) and (2.2) are the same. In other words the semantic model of the ET and corresponding BN are here similar and every CPT $p(X_i | X_{i-1}, \ldots, X_1)$ corresponds to the set of conditional mass functions given in generation i of the ET. The reverse translation under these circumstances is straightforward: The structure of the generations is clear and the conditional probabilities assigned to edges between generation $i-1$ and i are given from the CPT for $p(X_i | X_{i-1}, \ldots, X_1)$. Similarly work risk assessment considerations. To construct an $F-N$ curve using this BN, we only need to compute the joint mass function for $\{X_1, X_2, \ldots, X_n\}$ by applying Equation 2.1.

The application of Algorithm 1 to Example 1.7 from Chapter 1 gives the following.

Example 2.4. *Suppose we want to transform the ET from Example 1.7 into a BN using the generic Algorithm 1. The random variables in question are $X_1 := S \in \{1,2\}$, $X_2 := F \in \{y,n\}$, $X_3 := E \in \{n,s,l\}$. After the steps for $i = 1,2,3$ have been carried out, we obtain:*

1. *The final node set $V = \{X_1, X_2, X_3\}$.*
2. *The final edge set $E = \{(X_1, X_2), (X_1, X_3), (X_2, X_3)\}$, where (X_i, X_j) stands for a directed edge from X_i to X_j.*

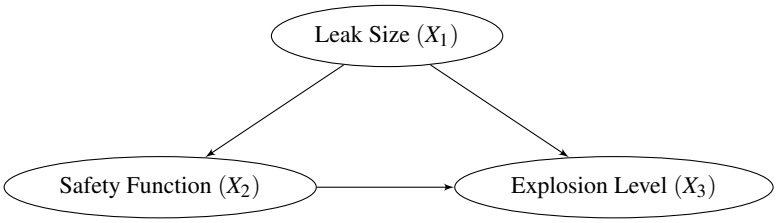

Figure 2-1: *Generic structure of a translated Bayesian network from Example 1.7.*

3. *We need to specify (at least) the following probabilities to set up the CPTs for all variables:*

$$
\begin{aligned}
&p(X_1 = 1),\\
&p(X_2 = 1|X_1 = 1), p(X_2 = 1|X_1 = 2),\\
&p(X_3 = 1|X_2 = 1, X_1 = 1), p(X_3 = 2|X_2 = 1, X_1 = 1),\\
&p(X_3 = 1|X_2 = 1, X_1 = 2), p(X_3 = 2|X_2 = 1, X_1 = 2),\\
&p(X_3 = 1|X_2 = 2, X_1 = 1), p(X_3 = 2|X_2 = 2, X_1 = 1),\\
&p(X_3 = 1|X_2 = 2, X_1 = 2), p(X_3 = 2|X_2 = 2, X_1 = 2).
\end{aligned}
$$

The resulting BN structure is visualised in Figure 2-1 and can be compared to the ET from Section 1.2. We do not display any CPTs here.

It is easy to see why this translation is far from useful. Generalising, we observe the following: Given a node X representing a discrete random variable with s many possible values and a parent set of size D, that is, $|\mathrm{pa}(X)| = D$, where each parent $X_{\mathrm{pa}_m} \in \mathrm{pa}(X)$, $m = 1, 2, \ldots, D$ has $|\mathscr{X}_{\mathrm{pa}_m}|$ possible values, one needs to specify $(s-1)\prod_{j=1}^{D} |\mathscr{X}_{\mathrm{pa}_m}|$ many probabilities for the (CPT) of X. This implies that the CPTs of the BN will grow rapidly if no checks for conditional independencies have been carried out. One of the main strengths of BNs lies just in the fact that they represent conditional independencies in their structure explicitly so that a joint distribution can be described by necessary building blocks only.

We will now focus on how to improve this trivial algorithm to incorporate simplifications for the resulting networks and hence potentially reduce the representation (both in terms of edges and parameter specifications) dramatically.

The idea to translate ETs into BNs is not new. (Marsh and Bearfield, 2008) describe and demonstrate the main idea of casting ETs into BNs while including a simple check of conditional independencies. They state "A causal arc to an event e_t [event here means a random variable of the ET] from an earlier event e_f is only needed if the probabilities labelling branches for event e_t depend on the outcome of event e_f. This can be seen in the event tree" and "More generally, if for all outcomes of e_t the probability $p(e_t|\ldots, e_f, \ldots)$ does not depend on the outcome of e_f (given the outcome of the other events), then the two events are 'conditionally independent' and the arc from e_f to e_t is not needed.". They apply this method to a case of derailment accidents analysis in

which they demonstrate the flexibility of BNs to represent a generalised risk model that includes the information of otherwise several separated location-specific ETs.

Remark 2.5. *We should note that (Marsh and Bearfield, 2008) use a reduced tree representation, i.e. they work with asymmetric trees. If the final consequence does not depend on the outcome for one of the random variables, then the corresponding branch is collapsed, a so called 'don't care' condition. We, on the other hand, constructed ETs in a symmetric way, such that all states of a variable are represented in each generation, even if some of the outcomes are impossible given the states of preceding variables. This is accounted for by assigning probability zero to edges associated with impossibilities. By doing that, representation and implementation become easier for our undertakings.*

In (Unnikrishnan et al., 2014) some advantages of BNs over ETs are demonstrated using simple examples of ET-to-BN translations with safety risk background. This includes easier backwards inference, better visualisation of structure and flexibility in adding new causal factors.

(Bobbioa et al., 2001) consider Fault Trees (FTs) as their starting point to form a BN and emphasise again the advantages of BNs both in modelling and inference. We remind that FTs usually precede ETs in a so called bow-tie approach to risk analysis. FTs have a tree-like structure, usually only containing binary variables and examine sources for potential failures that lead to some kind of accident whose consequences are analysed in a connected ET (thus forming a bow-tie looking graph).

(Khakzad et al., 2013) show in a similar fashion as the previous articles how a 'complete bow-tie' can be mapped into a BN and apply it to a case study of vapour ignition.

We propose an extension of Algorithm 1 and the above literature by using an information measure to identify conditional independencies or 'weak dependencies' in order to simplify the set of parents for each network node. To quantify the information losses when removing variables from a parent set, we need some notions from information theory. Those will be introduced in the next section.

2.2 Tools from information theory

Consider the following example to get an idea what a 'weak dependence' can look like.

Example 2.6. *(Similar tree branches)*
Suppose the following two branches make up the upper and lower part of the first / second generation of an ET having only binary variables, such that $\{1,2\}$ *is the set of values for a variable* X_1 *represented by the first generation and* $\{11,12\}$ *is the set of values for variable* X_2 *represented by the second generation. Figure 2-2 below shows this setup together with some conditional probabilities.*

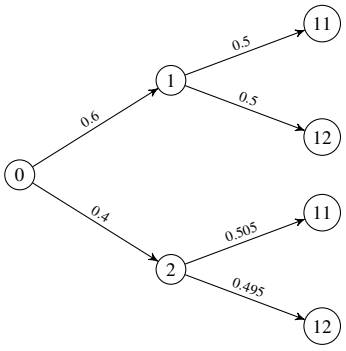

Figure 2-2: *Two binary tree branches with similar probabilistic structure.*

Clearly $p_{X_2|X_1}(11|1) \neq p_{X_2|X_1}(11|2)$ *and* $p_{X_2|X_1}(12|1) \neq p_{X_2|X_1}(12|2)$*, so that the two branches do not encode the conditional independence* $X_2 \perp X_1|\emptyset$*. However, we might regard them as 'almost' conditional independence as* $p_{X_2|X_1}(11|1) \approx p_{X_2|X_1}(11|2)$ *and* $p_{X_2|X_1}(12|1) \approx p_{X_2|X_1}(12|2)$ *and one could choose to only use one branch as approximation for both of them. Whether or not this is an acceptable approximation depends on the measure of dependence used and the application in mind.*

To measure conditional dependencies, we will repeatedly compare the induced conditional mass functions of the type $p(X_i|\mathbf{X_s} = \mathbf{x_s})$ to mass functions of the type $p(X_i|\mathbf{X_r} = \mathbf{x_r})$ for some instantiations $\mathbf{X_s} = \mathbf{x_s}$ and $\mathbf{X_r} = \mathbf{x_r}$ and where $\mathbf{X_r} \subset \mathbf{X_s} \subseteq \{X_1, X_2, \ldots, X_{i-1}\}$. If the comparison shows the equality of $p(X_i|\mathbf{X_s} = \mathbf{x_s})$ and $p(X_i|\mathbf{X_r} = \mathbf{x_r})$ for all possible choices of $\mathbf{x_s} \in \mathscr{X}_s$, we have identified the conditional independence $X_i \perp \mathbf{X_{s \setminus r}}|\mathbf{X_r}$.

Suppose the comparison shows a difference between the conditional distributions for at least two choices of instantiations of $\mathbf{X_s}$ that only differ on $\mathbf{X_{s \setminus r}}$; then we have identified a dependence and we have to decide how to aggregate these differences together to obtain a quantity that describes the strength of this dependence. We chose expectation as a function to aggregate these differences; this will turn out to be a natural choice. By using an information (loss) measure L that that quantifies the distributional dis-similarity of two induced conditional mass functions, a conditional independence such as $X_i \perp \mathbf{X_{s \setminus r}}|\mathbf{X_r}$ will arise whenever the expectation of the information loss L with respect to the joint distribution $P_{\mathbf{X_s}}$ of the $\mathbf{X_s}$ will satisfy

$$\mathbb{E}_{P_{\mathbf{X_s}}} L\left[p(X_i|\mathbf{X_s} = \mathbf{x_s}), p(X_i|\mathbf{X_r} = \mathbf{x_r})\right] = 0.$$

This will be a signification of 'the state of X_i does not depend on the state of $\mathbf{X_{s \setminus r}}$, given $\mathbf{X_r}$'. Then we might also think of a 'weak' conditional dependence whenever

$$\mathbb{E}_{P_{\mathbf{X_s}}} L\left[p(X_i|\mathbf{X_s} = \mathbf{x_s}), p(X_i|\mathbf{X_r} = \mathbf{x_r})\right] \leq \alpha,$$

for some small threshold α and could choose the approximation

$$p(X_i|\mathbf{X_s}) \approx p(X_i|\mathbf{X_r}).$$

In order to talk about the information loss when approximating one conditional distribution by another, we should first be able to measure the information content of one distribution, since each instantiation $\mathbf{X_s} = \mathbf{x_s}$ induces conditional probability functions $p(X_i|\mathbf{X_s} = \mathbf{s_s})$ and $p(X_i|\mathbf{X_r} = \mathbf{x_r})$. Let us introduce some notions from information theory that will be useful to that aim.

2.2.1 Entropy concept for probability distributions

Intuitively speaking, the information contained in a random variable (or its distribution) should be expressed by the amount of 'surprise' it encodes or in other words the uncertainty of the value it will take. By this we mean that if a random variable has its mass distributed in a lumpy rather than flat way, there is less uncertainty in its state. In the extreme case of a point mass, there is no surprise in the variable's state at all. The other extreme is a random variable with uniform distribution for which there is no tendency at all and every observation should be equally 'information-rich'.

There have been several properties suggested that a useful measure H of information / uncertainty of a distribution should exhibit. (Csiszár, 2008) gives an overview of the axiomatic characterisation of generalised information measures. The most famous paper in the area of information theory is probably (Shannon, 1948). He works with discrete distributions and proves the following theorem that determines the form of an information measure given some reasonable properties are fulfilled. (We have altered the notation and description of properties slightly to decouple the theorem from the context of the paper.)

Theorem 2.7. *(Uniqueness theorem)*
Suppose $H_n : \Delta_n \to \mathbb{R}_0^+$ is a function on the set of n-dimensional distributions

$$\Delta_n := \left\{ (p_1, \ldots, p_n) : p_i \geq 0, \sum_{i=1}^{n} p_i = 1 \right\}$$

for $n > 1$, that satisfies the following properties:
(i) H_n is continuous in the p_i.
(ii) Given P_n is the uniform distribution, i.e. $p_i = \frac{1}{n}$ for $i = 1, \ldots, n$, $H_n(P_n)$ is a monotonic increasing function of n.
(iii) $H_n(p_1, p_2, \ldots, p_n) = H_{n-1}(p_1 + p_2, p_3, \ldots, p_n) + (p_1 + p_2) H_2\left(\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2}\right)$.

The only type of function H_n, $n > 1$ satisfying the three above assumptions is of the form:

$$H_n(p_1, p_2, \ldots, p_n) = -K \sum_{i=1}^{n} p_i \log p_i,$$

where K is a positive constant.

For the special case $K = 1$, we recover the Shannon-entropy which has been used extensively for various applications. We will assume from now on that H means the Shannon-entropy. The dependence on n is usually clear from the context and we do not need to keep writing it. The following definitions and properties can be found in many texts, such as (Cover and Thomas, 2006).

Definition 2.8. *(Shannon-entropy)*
Let X be a discrete random variable with distribution P and values in the finite set $\mathscr{X} \subset \mathbb{R}$ and mass function p_X. The entropy $H(X)$ of X is defined as

$$H(X) = -\sum_{x \in \mathscr{X}} p_X(x) \log p_X(x),$$

with the convention $0 \cdot \log 0 = 0$. For a general random variable X with distribution P, we define

$$H(X) = -\mathbb{E}_P \log dP,$$

whenever this expression exists. We might also write $H(p_X)$ or $H(P_X)$ if the context is clear.

Similarly, for two (or more) random variables, say X and Y, one can define the joint entropy of Y and X and the conditional entropy of Y given X.

Definition 2.9. *(Joint entropy, conditional entropy)*
Let X, Y be discrete random variables with values in the finite sets $\mathscr{X}, \mathscr{Y} \subset \mathbb{R}$ and joint mass function $p_{Y,X}$. Then the joint entropy $H(Y,X)$ is defined as

$$H(Y,X) = -\sum_{x \in \mathscr{X}} \sum_{y \in \mathscr{Y}} p_{Y,X}(y,x) \log p_{Y,X}(y,x).$$

The conditional entropy of Y, given X (which specifies the amount of uncertainty about Y, given the state of X) is then defined by

$$H(Y|X) = -\sum_{x \in \mathscr{X}} p_X(x) \sum_{y \in \mathscr{Y}} p_{Y|X}(y|x) \log p_{Y|X}(y|x).$$

More generally, if X, Y are random variables with joint distribution $P_{X,Y}$, we define

$$H(Y,X) = -\mathbb{E}_{P_{X,Y}} \log dP_{X,Y}.$$

The following properties are convenient to relate the different quantities later on, see again (Cover and Thomas, 2006) for example for more information.

Lemma 2.10. *(Some properties of the entropy H)*
Let X, Y be discrete random variables as above. Then the following properties hold:
(i) $H(X) \geq 0$,

(ii) $H(Y,X) \leq H(Y)+H(X)$,
(iii) $H(Y|X) = H(Y,X)-H(X)$.

Remark 2.11. *For the general case of arbitrary random variables X,Y, in order to avoid difficulties one can also decide to define $H(Y|X) := H(Y,X)-H(X)$.*

There have been different suggestions for generalisations of the entropy concept; usually by relaxing some of the defining properties. One more general class of entropies that include the Shannon-entropy as a limit case are the so called Rényi-entropies (Rényi, 1961). Such entropies allow to emphasise on different aspects of distributions.

In this work we will mainly compare conditional entropies because conditional distributions are a crucial part of the translation and simplification algorithm. The last notion we define here is conditional mutual information (CMI) which expresses the informational difference between two conditional distributions.

Definition 2.12. *(Conditional mutual information)*
Let X,Y,Z be discrete random variables or sets of random variables. We define the conditional mutual information $CMI(X,Y|Z)$ between X and Y, given Z to be

$$MI(X,Y|Z) = H(X|Z)-H(X|Y,Z).$$

This definition extends to general random variables whenever the conditional entropies exist.

Now that we have defined some well known information measures, we can look at how to compare the information divergence between two distributions.

2.2.2 Information divergences to compare probability distributions

There are many different approaches to compare probability distributions; a popular divergence measure is the so called Kullback-Leibler divergence (KLD).

Definition 2.13. *(Kullback-Leibler divergence / relative entropy)*
Let P and Q be two discrete probability distributions with common support $\mathscr{X} = \{x_1,x_2,\ldots,x_n\}$. Let $p : \mathscr{X} \to [0,1]$ and $q : \mathscr{X} \to [0,1]$ be the two probability mass functions for P and Q. Then

$$D(P\|Q) = D(p\|q) := \sum_{x\in\mathscr{X}} p(x)\log\frac{p(x)}{q(x)}.$$

We can define $D(p\|q) = +\infty$ if $q(x) = 0$ for some $x \in \mathscr{X}$ with $p(x) > 0$.

More generally, this concept can be defined as follows. Let P and Q be probability measures on $(\Omega,\mathscr{A})$ such that $P \ll Q$. Then

$$D(P\|Q) := \int dP \log\frac{dP}{dQ} = \mathbb{E}_P\left[\log\frac{dP}{dQ}\right].$$

If $P \ll Q$, one can define $D(P\|Q) = +\infty$.

If X and Y are random variables distributed according to P_X and Q_Y respectively, then we can also write $D(X\|Y)$ to mean $D(P_X\|Q_Y)$.

Remark 2.14. *We note here that the KLD is not symmetric in its arguments and hence not a mathematical distance in the usual sense. The KLD is a useful information measure that can be interpreted as the 'information loss' when approximating its left argument distribution by its right argument distribution.*

Remark 2.15. *It is possible to express the KLD as the difference $D(P\|Q) = H(P) - H_P(Q)$, where $H_P(Q) := -\sum_{x\in\mathscr{X}} p(x)\log q(x)$ (in general: $H_P(Q) := \mathbb{E}_P[-\log dQ]$) is the so called cross-entropy. The interpretation of D as the information loss when approximating one distribution by another comes among other things, from this relation.*

Typical values of the KLD may be hard to interpret but one can think in terms of a biased coin to grasp the magnitude of information difference. We present a result that guides intuition about the size of KLD which we discovered as an exercise from (Koski and Noble, 2009):

Proposition 2.16. *(Calibration of Kullback-Leibler divergence)*
Let $D(P\|Q) = k$ be the value of the Kullback-Leibler divergence between two probability distributions defined on $\mathscr{X} = \{x_1,\ldots,x_n\}$. Then $D(P\|Q)$ is the same as between a fair Bernoulli distribution (denoted $B\left(\frac{1}{2}\right)$) and a Bernoulli distribution $B(h(k))$, where $h(k) = \frac{1}{2}\left(1 \pm \sqrt{1-e^{-2k}}\right)$.

Proof. A Bernoulli distribution with parameter b on $\{0,1\}$ has mass b on $\{1\}$ and $1-b$ on $\{0\}$. Hence

$$\begin{aligned} D\left(B\left(\frac{1}{2}\right)\middle\|B(h(k))\right) &= \frac{1}{2}\log\frac{\frac{1}{2}}{1-\frac{1}{2}\left(1\pm\sqrt{1-e^{-2k}}\right)} + \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{1}{2}\left(1\pm\sqrt{1-e^{-2k}}\right)} \\ &= \frac{1}{2}\log\frac{\frac{1}{4}}{\frac{1}{2}\left(1\pm\sqrt{1-e^{-2k}}\right) - \frac{1}{4}\left(1\pm\sqrt{1-e^{-2k}}\right)^2} \\ &= k. \end{aligned}$$

□

A graph of the function $h(k)$ can be seen in Figure 2-3.

The KLD is part of a more general class of divergence functions between two probability measures. The f-divergences, introduced by among others from (Ali and Silvey, 1966), are defined next. This class of functions allows to relate the KLD to other types of divergences.

Definition 2.17. *(f-divergence)*
Let $f : (0,\infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$ and suppose P, Q are discrete probability measures with mass functions p and q. Then we define

$$D_f(P\|Q) := \sum_{x\in\mathscr{X}} f\left(\frac{p(x)}{q(x)}\right) q(x). \tag{2.3}$$

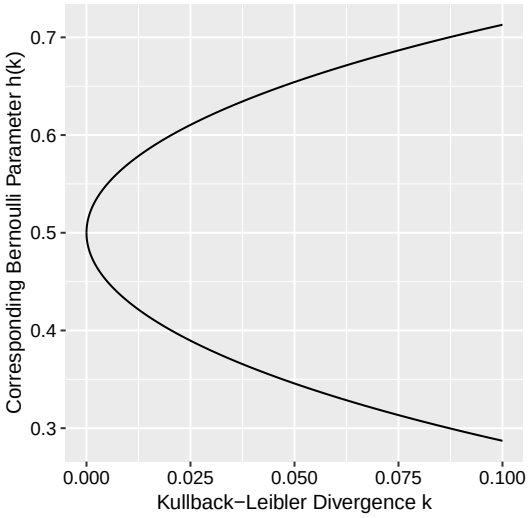

Figure 2-3: *Comparison of the Kullback-Leibler divergence to the difference between a fair coin and an unfair coin.*

In a more general setting, given P,Q are any kind of probability measures such that $P \ll Q$, the f-divergence from P to Q is defined as follows:

$$D_f(P\|Q) := \int f\left(\frac{dP}{dQ}\right) dQ. \tag{2.4}$$

The Kullback-Leibler divergence is the special case of $f(t) = t\log t$. The class of f-divergences has several natural properties that are desirable for a measure of 'dis-similarity'. We note the following two can be found in (Sason and Verdú, 2016) for example, and are proven using Jensen's inequality (Theorem 2.30).

Theorem 2.18.
For the f-Divergence from P to Q it holds

$$D_f(P\|Q) \geq 0,$$

and

$$D_f(P\|Q) = 0 \Leftrightarrow P = Q.$$

This theorem allows us to establish distributional equality by computing the Kullback-Leibler divergence. We have equality exactly when the KLD is vanishing. Now we introduced all necessary notions and can return and state the proposed algorithm.

2.3 A translation algorithm based on an information measure

In this section we present the proposed sequential algorithm to translate an ET to a BN using the Kullback-Leibler divergence as a measure of similarity.

We will start with an empty network and in each step i of the algorithm, we add a new variable X_i to the already created network and initially this newly added variable X_i is assumed to have a full set of potential parents $\mathrm{pa}_{\mathrm{pot}}(X_i) := \{X_1, \ldots, X_{i-1}\}$. The aim of the current step is then to find a smaller subset $\mathrm{ess}(X_i) \subseteq \mathrm{pa}_{\mathrm{pot}}(X_i)$, such that the information loss when approximating $p(X_i|X_{i-1}, \ldots, X_1)$ by $p(X_i|\mathrm{ess}(X_i))$ stays below some threshold α.

The suggested algorithm executes a fairly large search for each step (similar to a greedy search), but we impose a stopping criterion that usually will be met quickly in applications. The general description of the idea can be found in Algorithm 2; we note that the crucial difference to the generic translation lies in Step 4.

Algorithm 2 Translation of an ET to a BN using an information measure.

Input: Event Tree with n generations representing variables $X_1, X_2, \ldots, X_n$, with ordering σ given from the sequence of generations and all conditional probabilities along the edges. Threshold α.
Output: Translated, potentially simplified Bayesian network.

Let $V := \{\}$ and $E := \{\}$.
For $i = 1, \ldots, n$ do:

1. Create a node X_i and update the node set $V \leftarrow V \cup \{X_i\}$.
2. Assign an initial set of potential parents $\mathrm{pa_{pot}(X_i)} := \{X_{i-1}, \ldots, X_1\}$.
3. Initialise a set of essential parents: $\mathrm{ess}(X_i) := \{\}$.
4. While $\left(\mathrm{pa}_{\mathrm{pot}}(X_i) \setminus \mathrm{ess}(X_i) \neq \emptyset\right)$ do:
   (a) For each $X_k \in \mathrm{pa}_{\mathrm{pot}}(X_i) \setminus \mathrm{ess}(X_i)$ do:
      i. Compute the magnitude α_k of the expected information loss when approximating $p(X_i|\mathrm{pa}_{\mathrm{pot}}(X_i))$ by $p(X_i|\mathrm{pa}_{\mathrm{pot}}(X_i) \setminus X_k)$.
      ii. If $\alpha_k > \alpha$: Set $\mathrm{ess}(X_i) \leftarrow \mathrm{ess}(X_i) \cup X_k$.
   (b) If $(\{\alpha_k : \alpha_k \leq \alpha\} \neq \emptyset)$: Remove the $X_{\tilde{k}}$ for which $\tilde{k} = \mathrm{argmin}_k\{\alpha_k : \alpha_k \leq \alpha\}$ from $\mathrm{pa}_{\mathrm{pot}}(X_i)$:
   $$\mathrm{pa}_{\mathrm{pot}}(X_i) \leftarrow \mathrm{pa}_{\mathrm{pot}}(X_i) \setminus X_{\tilde{k}}.$$
   Else: Stop and go to step 5.
   (c) Adjust the threshold α to account for the information loss by the removal of X_k. $\alpha \leftarrow g(\alpha, \alpha_{\tilde{k}})$.
5. Define the parent set $\mathrm{pa}(X_i) := \mathrm{ess}(X_i)$ and update the edge set
   $$E \leftarrow E \cup \bigcup_{j:X_j \in \mathrm{pa}(X_i)} \{(X_j, X_i)\}.$$
6. Define the CPT for $p(X_i|\mathrm{pa}(X_i))$ by using the available information.

A number of remarks are necessary at this point.

Remark 2.19.

(i) The ordering σ of the variables $\{X_1, X_2, \ldots, X_n\}$ is not fundamental for the correct functioning of the algorithm, even though different orderings will most likely result in different BNs. We consider the 'natural' order to be given by the generations of the ET. ETs often describe a physical or timely progression of events. Furthermore by using this ordering we can readily use all given conditional mass functions from the ET data. If we choose a different ordering we must potentially compute new conditional mass functions before even doing any simplifications.

(ii) The set of essential parents serves as a way of collecting all the variables that are deemed 'essential' to determine the outcome of X_i. If at one point in step i the removal of one of the candidate parents X_k in $\mathrm{pa}_{\mathrm{pot}}(X_i) \setminus ess(X_i)$ leads to a total information loss that is above threshold α, then X_k is 'essential'; it will definitely be kept in the parent set and does not need to be checked for possible removal later on.

(iii) If in step 4(b) there no unique $\tilde{k}$, then the one with smallest index value will be used.

(iv) The adjustment of the threshold in step 4(c) can be of different form (choices of function g). In the case where the information loss from removing variables X_l and X_m from the set of candidate parents is the same as the sum of individual information losses from removing first X_l and then X_m, it would be reasonable to set $\alpha \leftarrow \alpha - \alpha_{\tilde{k}}$, i.e. $g(\alpha, \alpha_{\tilde{k}}) := \alpha - \alpha_{\tilde{k}}$. We will motivate this choice in the next subsection.

(v) In order to compare $p(X_i|\mathbf{X_s})$ with $p(X_i|\mathbf{X_r})$ for $\mathbf{X_r} \subset \mathbf{X_s}$, we will need to 'marginalise out conditional variables' repetitively which is generally done by marginalising variables from joint distributions. In our implementation of the algorithm, we will use a fairly efficient way of computing conditional distributions via the R-package `gRain`.

In the last remark we hinted with (iii) at the situation where the individual information losses are additive. The expectation version of the introduced Kullback-Leibler divergence (and CMI) fulfils this property, as we will show in the next subsection. We will present the specifics when using the CMI as measure for information divergence and keep this choice for the rest of this chapter and the next.

2.3.1 Translation using conditional mutual information

For emphasise, we will change our notation for the next two subsections slightly. In step i in the algorithm, we call the newly added variable whose parent set we are trying to determine Y instead of X_i. The vector of variables obtained from going through iteration 1 to $i-1$ will be denoted by $\mathbf{X} = (X_1, \ldots, X_{i-1})$. The induced conditional mass function of Y, given $\mathbf{x_s} \subseteq \mathbf{x} \in \mathscr{X}$ will then be written as $p(y|\mathbf{x_s})$, etc.

We mentioned before that our aim is to measure the information loss to incur if we 'replace' a conditional mass function $p(y|\mathbf{x})$ with a conditional mass function $p(y|\mathbf{x_s})$ that is considered 'simpler' in the sense of $\mathbf{x_s} \subset \mathbf{x}$. A convenient choice to measure this is the expected Kullback-Leibler divergence. D generally depends on the instantiations of any conditioning variables $\mathbf{x}$ and

hence could be regarded as a function of $\mathbf{x}$. To aggregate these values for different choices of $\mathbf{x}$ by using the expectation leads to the CMI; it has been used in similar context for feature selection for a classification problem in (Koller and Sahami, 1996).

One can show for the discrete case that the expected Kullback-Leibler divergence between a conditional distribution with mass function $p(y|\mathbf{x})$ and simpler version with mass function $p(y|\mathbf{x_s})$ is equivalent to the difference between the conditional entropies $H(Y|\mathbf{X_s})$ and $H(Y|\mathbf{X})$. This is the same as the CMI between Y and $\mathbf{X}_{-\mathbf{s}}$, given $\mathbf{X_s}$:

$$\begin{aligned}
\mathbb{E}_{P_\mathbf{x}} D(p(y|\mathbf{x}) \| p(y|\mathbf{x_s})) &= \sum_{\mathbf{x}} p(\mathbf{x}) \sum_{y} p(y|\mathbf{x}) \log \frac{p(y|\mathbf{x})}{p(y|\mathbf{x_s})} \\
&= \sum_{\mathbf{x},y} p(y,\mathbf{x}) \log \frac{p(y|\mathbf{x})}{p(y|\mathbf{x_s})} \qquad (2.5) \\
&= \sum_{\mathbf{x},y} p(y,\mathbf{x}) \log p(y|\mathbf{x}) - \sum_{\mathbf{x},y} p(y,\mathbf{x}) \log p(y|\mathbf{x_s}) \\
&= \sum_{\mathbf{x},y} p(y,\mathbf{x}) \log p(y|\mathbf{x}) - \sum_{\mathbf{x_s},y} p(y,\mathbf{x_s}) \log p(y|\mathbf{x_s}) \\
&= -H(Y|\mathbf{X}) + H(Y|\mathbf{X_s}) \\
&= CMI(Y, \mathbf{X}_{-\mathbf{s}}|\mathbf{X_s}). \qquad (2.6)
\end{aligned}$$

Hence we can interpret the expected Kullback-Leibler divergence as the difference in the information needed to describe Y, given $\mathbf{X_s}$ and given $\mathbf{X}$.

Remark 2.20. *For our algorithm, form (2.5) is most suitable, as $p(y|\boldsymbol{x})$ is information directly given in the ET and $p(y,\boldsymbol{x})$ is obtained by simple multiplication of table columns. Only one set of marginalisation operations is required to obtain $p(y|\boldsymbol{x}_s)$.*

The basis to use the CMI to determine and quantify conditional dependencies are the following two facts.

Remark 2.21. *Theorem 2.18 implies that if we obtain $D(p(y|\mathbf{x}) \| p(y|\mathbf{x_s})) = 0$ for all $\mathbf{x}$ then $p(y|\mathbf{x}) = p(y|\mathbf{x_s})$ for all $\mathbf{x}$. This is equivalent to saying that Y is conditionally independent of $\boldsymbol{X}_{-\mathbf{s}}$, given $\mathbf{X_s}$. ($Y \perp \mathbf{X}_{-\mathbf{s}}|\mathbf{X_s}$)*

Corollary 2.22. *From Theorem 2.18 and Equation (2.6) we immediately obtain*

$$CMI(Y, \mathbf{X}_{-\mathbf{s}}|\mathbf{X_s}) \geq 0$$

and

$$CMI(Y, \mathbf{X}_{-\mathbf{s}}|\mathbf{X_s}) = 0 \Leftrightarrow Y \perp \mathbf{X}_{-\mathbf{s}}|\mathbf{X_s}.$$

The following results show some of the convenience of the Kullback-Leibler divergence for comparisons of the type we are interested in.

Proposition 2.23. *Let* $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \subseteq \mathbf{X}$*, such that* $\mathbf{X}_1 \subseteq \mathbf{X}_3$*,* $\mathbf{X}_2 \subseteq \mathbf{X}_3$*. The difference between two expected Kullback-Leibler divergences*

$$\mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] \text{ and } \mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right]$$

does not depend on the conditioning variables ($\mathbf{X}_3$*) of the reference distribution other than the conditioning variables contained on the right argument distributions (*$\mathbf{X}_2$ *and* $\mathbf{X}_1$*).*

Proof. We established that the expected Kullback-Leibler divergence can be written as the difference between two conditional entropies, hence:

$$\mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] = H\left(Y|\mathbf{X}_2\right) - H\left(Y|\mathbf{X}_3\right)$$

and

$$\mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right] = H\left(Y|\mathbf{X}_1\right) - H\left(Y|\mathbf{X}_3\right).$$

It follows

$$\begin{aligned}
&\mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] - \mathbb{E}_{P_{\mathbf{x}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right] \\
&\quad = H\left(Y|\mathbf{X}_2\right) - H\left(Y|\mathbf{X}_3\right) - \left(H\left(Y|\mathbf{X}_1\right) - H\left(Y|\mathbf{X}_3\right)\right) \\
&\quad = H\left(Y|\mathbf{X}_2\right) - H\left(Y|\mathbf{X}_1\right)
\end{aligned}$$

which is not dependent on variables in $\mathbf{X} \setminus \{\mathbf{X}_2 \cup \mathbf{X}_1\}$. □

The next corollary shows that if the removal of a number of variables from the full set of potential parents leads to a higher CMI than the removal of a different set of variables, then this relation is preserved even if we had started not with the full set of potential parents, but a smaller set. In other words, if removing a variable X_k from the full set of potential parents is 'worse' than removing X_m, then this is true, even if we had firstly removed X_l from the set of potential parents.

Corollary 2.24. *Let* $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \subseteq \mathbf{X}$*, such that* $\mathbf{X}_1 \subseteq \mathbf{X}_3$*,* $\mathbf{X}_2 \subseteq \mathbf{X}_3$*. Then*

$$\begin{aligned}
&\mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] \geq \mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right] \\
&\qquad\qquad \Leftrightarrow \\
&\mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_2\right)\right] \geq \mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_1\right)\right].
\end{aligned}$$

Proof. Note that

$$\begin{aligned}
&\mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] - \mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right] \\
&\qquad = H\left(Y|\mathbf{X}_2\right) - H\left(Y|\mathbf{X}_1\right) \\
&= \mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_2\right)\right] - \mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_1\right)\right].
\end{aligned}$$

Hence, if

$$\mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_2\right)\right] - \mathbb{E}_{P_{\mathbf{X}_3}} D\left[p\left(y|\mathbf{x}_3\right) \| \, p\left(y|\mathbf{x}_1\right)\right] \geq 0,$$

then

$$\mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_2\right)\right] - \mathbb{E}_{P_{\mathbf{X}}} D\left[p\left(y|\mathbf{x}\right) \| \, p\left(y|\mathbf{x}_1\right)\right] \geq 0$$

and vice versa. □

A very useful property of the conditional entropy is its additivity which will turn out to be crucial for our algorithm. It allows us to calculate the total difference in entropy when removing several variables from a candidate parent set as the sum of the effects of gradually removing variables one by one from the parent set.

Lemma 2.25. *Let* $\mathbf{X} \setminus \mathbf{X_s} = \{X_{\tilde{1}}, \ldots, X_{\tilde{r}}\}$ *and* $\mathbf{X}_0 := \{\}, \mathbf{X}_1 := \{X_{\tilde{1}}\}, \ldots, \mathbf{X}_r = \{X_{\tilde{1}}, \ldots, X_{\tilde{r}}\}$. *Then*

$$H(Y|\mathbf{X}) - H(Y|\mathbf{X_s}) = \sum_{j=0}^{r-1} \left[H(Y|\mathbf{X} \setminus \mathbf{X}_j) - H(Y|\mathbf{X} \setminus \mathbf{X}_{j+1})\right].$$

Proof. The right hand side presents a telescopic sum, such that

$$\sum_{j=0}^{r-1} \left[H(Y|\mathbf{X} \setminus \mathbf{X}_j) - H(Y|\mathbf{X} \setminus \mathbf{X}_{j+1})\right] = H(Y|\mathbf{X} \setminus \mathbf{X}_0) - H(Y|\mathbf{X} \setminus \mathbf{X}_r) = H(Y|\mathbf{X}) - H(Y|\mathbf{X_s}).$$

□

Algorithm 3 shows the determination of a parent set for each variable in step i, when using the CMI as a measure of similarity.

Remark 2.26. *We summarise some convenient properties of Algorithm 3.*
(i) The fact that the CMI can be written as difference between two conditional entropies together with Lemma 2.25 allows us to keep track of the total information loss as sum of individual information losses coming from individual removals of variables. This can be seen in Step 4(a)(i) and 4(c). In each iteration of determining a parent set, the threshold is adjusted by subtracting the loss occurred of removing a single variable and then used as 'new' threshold when testing if further potential parents can be removed.
(ii) Corollary 2.24 allows us to use the set of essential parents in the ways we do in Step 4, 4(a)(ii) and 5. Once the removal of a variable from the set of potential parents leads to an information loss larger than the current threshold α*, we do not need to test it for removal in a later iteration of Step 4 again. In any later step it will hurt the threshold as well. Thus we collect such variables in the set of essential parents. In case none of the potential parents can be removed, this would be noted immediately since all potential parents would move into the bag of essential parents and Step 4 would end in one iteration.*

The next subsection addresses an approach to the problem of how to chose the threshold α for the amount of acceptable information loss.

Algorithm 3 Determination of a simplified parent set for a variable Y.

Input: Distributions $p(Y|X_{i-1},\dots,X_1)$, $p(X_j|\text{pa}(X_j))$ for $j=1,\dots,i-1$.
Output: Parent set $\text{pa}(Y)$, such that $CMI(Y,\{\mathbf{X}\setminus\text{pa}(Y)\}|\text{pa}(Y))\leq\alpha$.

1. Assign an initial set of potential parents $\text{pa}_{\text{pot}}(Y):=\{X_{i-1},\dots,X_1\}$.
2. Initialise a set of essential parents: $\text{ess}(Y):=\{\}$.
3. Compute $\alpha_0:=H(Y|X_1,\dots,X_{i-1})$.
4. While $\left(\text{pa}_{\text{pot}}(Y)\setminus\text{ess}(Y)\neq\emptyset\right)$ do
   (a) For each $X_k\in\text{pa}_{\text{pot}}(Y)\setminus\text{ess}(Y)$ do:
      i. Compute $\alpha_k:=H(Y|\text{pa}_{\text{pot}}(Y)\setminus X_k)$.
      ii. If $\alpha_k>\alpha_0$: $\text{ess}(Y)\leftarrow\text{ess}(Y)\cup X_k$.
   (b) If $(\{\alpha_k:\alpha_k-\alpha_0\leq\alpha\}\neq\emptyset)$: Let $\tilde{\alpha}:=\min_k\{\alpha_k:\alpha_k-\alpha_0\leq\alpha\}$ and remove the $X_{\tilde{k}}$ for which $\tilde{k}=\text{argmin}_k\{\alpha_k:\alpha_k-\alpha_0\leq\alpha\}$ from $\text{pa}_{\text{pot}}(Y)$:
   $$\text{pa}_{\text{pot}}(Y)\leftarrow\text{pa}_{\text{pot}}(Y)\setminus X_{\tilde{k}}.$$
   Else: Stop and go to step 5.
   (c) Set $\alpha_0\leftarrow\tilde{\alpha}$ and $\alpha\leftarrow\alpha-\tilde{\alpha}$.
5. Set $\text{pa}(Y)\leftarrow\text{ess}(Y)$.

2.3.2 The choice of a suitable threshold α

The expected Kullback-Leibler divergence (or CMI) is not relating to intuition in a straightforward way. What does it mean if we obtain a CMI value of, say, 0.001? Does that translate into an approximation with good quality? Questions as such are necessary in order to decide how an appropriate threshold α for the translation should be chosen in practice. In this section we develop some inequalities that will relate the CMI to more intuitively accessible divergences within our context.

The total variation between two probability measures can be regarded as a worst case difference and should be intuitively more appealing to a practitioner. It can also be recovered as an f-divergence for the choice $f(t)=\frac{1}{2}|t-1|$.

Definition 2.27. *(Total variation distance)*
We define the total variation distance $\delta(P,Q)$ between two measures P,Q on $(\mathscr{X},\mathscr{A})$ to be

$$\delta(P,Q):=\sup_{A\in\mathscr{A}}|P(A)-Q(A)|.$$

Hence $\delta(P,Q)$ can be interpreted as the 'largest difference' between the event probabilities with respect to P and Q. For the case where $|\mathscr{X}|<\infty$, which is often true in applications, it holds $\{a:a\in\mathscr{X}\}\subseteq\{A:A\in\mathscr{A}\}$ and hence

$$\max_{a\in\mathscr{X}}\{|P(a)-Q(a)|\}\leq\sup_{A\in\mathscr{A}}\{|P(A)-Q(A)|\}.$$

Thus, the maximum difference between the probabilities of any two points is also not larger than the total variation distance.

We present a modified version of the general result from (Gilardoni, 2010), such that we can relate δ to f-Divergences:

Theorem 2.28. *(Pinsker's type inequality for f-Divergences)*
Assume that $f(u)$ is a convex function, differentiable up to order three at $u = 1$ and such that $f''(1) > 0$. Suppose furthermore that the following inequality holds.

$$\tilde{f}(u)\left[1 + (1 - w_f)(u-1)\right] \geq \frac{f''(1)}{2}(u-1)^2,$$

where $w_f = 1 + \frac{f'''(1)}{3f''(1)}$ and $\tilde{f}(u) = f(u) - f'(1)(u-1)$, where it is assumed that $\tilde{f}(u) > 0$, whenever $u \neq 1$. Then the inequality

$$2f''(1)\delta(P,Q)^2 \leq D_f(P\|Q) \tag{2.7}$$

holds for any two measures P, Q.

For $f(t) = t\log t$, we obtain the following well known special case, the proof of which can be found in many sources such as (Tsybakov, 2009).

Corollary 2.29. *(Pinsker's inquality)*

$$\delta(P,Q) \leq \sqrt{\frac{1}{2}D(P\|Q)}.$$

These results allow us to find a reasonable information loss threshold α in terms of the total variation distance. Based on Corollary 2.29 we deduct the following inequality for the translation algorithm. For each fixed instantiation of conditioning variables $\mathbf{x}$, we can bound the 'local maximum difference' between two choices of parent sets for the variable Y:

$$m(y,\mathbf{x},\mathbf{x_s}) := \max_{y\in\mathscr{Y}} |p(y|\mathbf{x}) - p(y|\mathbf{x_s})| \leq \sqrt{\frac{1}{2}D(p(y|\mathbf{x})\,\|\,p(y|\mathbf{x_s}))}. \tag{2.8}$$

One can employ Jensen's inequality, which we state below, to then reach an expression that relates to CMI.

Theorem 2.30. *(Jensen's inequality)*
Let X be a real-valued random variable with distribution P_X and let ϕ be a convex function. Then

$$\phi(\mathbb{E}_{P_X}(X)) \leq \mathbb{E}_{P_X}(\phi(X)).$$

Remark 2.31. *From now on all expectations are understood to be with respect to P_X or $P_{\mathbf{X}}$. This should always be clear from the context and we suppress to write the measure with any expressions involving the expectation. Should we need to emphasise that expectation for a random variable X with distribution P and mass function p_X is taken, then this can also be written as $\mathbb{E}_{p_X}$.*

We take the expectation on expression (2.8) (with respect to $P_{\mathbf{x}}$) and then apply Jensen's inequality on the left side to obtain the following:

$$2\left(\mathbb{E}m\left(y,\mathbf{x},\mathbf{x_s}\right)\right)^2 \leq \mathbb{E}2(m\left(y,\mathbf{x},\mathbf{x_s}\right))^2 \leq \mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right). \tag{2.9}$$

A more accessible way of deciding a threshold α could hence be guided by the inequality

$$\mathbb{E}m\left(y,\mathbf{x},\mathbf{x_s}\right) \leq \sqrt{\frac{1}{2}\mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right)}, \tag{2.10}$$

so that a choice of the 'expected maximum difference' $\mathbb{E}m\left(y,\mathbf{x},\mathbf{x_s}\right)$ can be made and the bound

$$\mathbb{E}m\left(y,\mathbf{x},\mathbf{x_s}\right) \leq \alpha,$$

is fulfilled whenever

$$\mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right) \leq 2\alpha^2.$$

A slightly different and somewhat more conservative way to chose the threshold α is based on the application of Markov's inequality. This approach will relate α to the probability of having a maximum difference larger than a chosen value.

Theorem 2.32. *(Markov's inequality)*
Let X be a real-valued random variable and ϕ be a monotonically increasing non-negative function on $[0,\infty)$. For any $a > 0$, such that $\phi(a) > 0$ we have

$$\mathbb{P}\left(|X| \geq a\right) \leq \frac{\mathbb{E}(\phi(|X|))}{\phi(a)}.$$

Combining the right-hand inequality of (2.9) with Markov's inequality, one obtains

$$2a\mathbb{P}\left(m\left(y,\mathbf{x},\mathbf{x_s}\right)^2 \geq a\right) \leq \mathbb{E}2(m\left(y,\mathbf{x},\mathbf{x_s}\right))^2 \leq \mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right). \tag{2.11}$$

Hence, if one uses Equation 2.11 and wants to ensure that $\mathbb{P}\left(m\left(y,\mathbf{x},\mathbf{x_s}\right) \geq c\right) \leq \alpha$ for some $c > 0$, then one should chose $2\alpha c^2$ as an upper threshold for $\mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right)$. If

$$\mathbb{E}D\left(p\left(y|\mathbf{x}\right)\|p\left(y|\mathbf{x_s}\right)\right) \leq 2\alpha c^2,$$

then

$$2c^2\mathbb{P}\left(m\left(y,\mathbf{x},\mathbf{x_s}\right) \geq c\right) \leq 2\alpha c^2.$$

This method is more conservative in the sense that Markov's inequality is not sharp.

Remark 2.33. *We have derived some upper bounds on the maximum difference between two 'local' conditional probabilities. Pinsker's inequality was well suited for this task and we are able to force this local maximum difference towards zero as we choose lower thresholds. In some cases*

practitioners are interested to bound the ratio of probabilities and their approximation instead. We will meet an example in Chapter 4 which highlights why bounding differences might not always be adequate. There are some other information measures, such as the Chain-Darwiche measure (Chan and Darwiche, 2005), which could also be used in the framework of Algorithm 2 and relates more closely to ratios of probabilities.

2.3.3 The connection between local error bound and global approximation

The KLD as a measure of information has the property that it connects local and global information in a simple way. Generally, as (Tong and Koller, 2001) mention, the KLD decomposes with the graphical structures of BNs. Suppose $P_{\mathbf{X}}$ and $Q_{\mathbf{X}}$ are two joint distributions over the variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ as before with joint mass functions $p(\mathbf{x})$ and $q(\mathbf{x})$. Suppose that they can be factorised as

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\text{pa}(X_i))$$

and

$$Q(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} Q(X_i|\text{pa}(X_i)),$$

then

$$D\left[P(X_1, \ldots, X_n) \| Q(X_1, \ldots, X_n)\right] = D\left[\prod_{i=1}^{n} p(x_i|\text{pa}(x_i)) \middle\| \prod_{i=1}^{n} q(x_i|\text{pa}(x_i))\right] \tag{2.12}$$

$$= \sum_{i=1}^{n} \mathbb{E}_{P_{\text{pa}(X_i)}} D\left[p\left(x_i|\text{pa}(x_i)\right) \| q\left(x_i|\text{pa}\left(x_i\right)\right)\right]. \tag{2.13}$$

Equation 2.13 can be interpreted as: the 'global' information divergence is equal to the sum of 'local' information divergences. We will see that this equality only holds approximately with our algorithm. Let us walk through the sequence of steps where for the joint distribution P, $\text{pa}_P(X_i) = \{X_1, X_2, \ldots, X_{i-1}\}$ and we want to determine in step i a suitable parent set $\text{pa}_Q(X_i) \subseteq \text{pa}_P(X_i)$ for X_i in Q.

Suppose $i = 1$. No simplifications are carried out and we let $q(x_1) = p(x_1)$. For $i = 2$ we compare $p(x_2|x_1)$ with $p(x_2)$. If $CMI(X_2, X_1|\emptyset) \leq \alpha$ we set $q(x_2|\text{pa}(x_2)) = p(x_2)$ and else $q(x_2|\text{pa}(x_2)) = p(x_2|x_1)$. As we proceed to $i = 3$, the strive for a balance of computational ease and good approximation lead to the idea to employ the approximate distribution $q(x_2|\text{pa}(x_2))$ as a substitute for $p(x_2|\text{pa}(x_2))$. This can be regarded a natural idea if one builds up the network sequentially and only keeps what has been translated and simplified so far.

In step i of the algorithm, to do more comparisons, the joint mass functions of $p(x_i, x_{i-1}, \ldots, x_1)$ and $q(x_i, x_{i-1}, \ldots, x_1)$ are needed. Here, if only the already simplified mass functions are kept, we use the approximation

$$q(x_i, x_{i-1}, \ldots, x_1) \approx p(x_i|x_{i-1}, x_{i-2}, \ldots, x_1) q(x_{i-1}, x_{i-2}, \ldots, x_1).$$

This means we really compute $\mathbb{E}_{q_{(x_1,\ldots,x_{i-1})}} D\left[p\left(x_i|x_{i-1},\ldots,x_1\right)\|q\left(x_i|\mathrm{pa}\left(x_i\right)\right)\right]$, where the expectation is with respect to the already simplified distribution. It may be critiqued whether

$$\mathbb{E}_{q_{(x_1,\ldots,x_{i-1})}} D\left[p\left(x_i|x_{i-1},\ldots,x_1\right)\|q\left(x_i|\mathrm{pa}\left(x_i\right)\right)\right] \approx \mathbb{E}_{p_{(x_1,\ldots,x_{i-1})}} D\left[p\left(x_i|x_{i-1},\ldots,x_1\right)\|q\left(x_i|\mathrm{pa}\left(x_i\right)\right)\right]$$

provides a good approximation at all times.

(Minka, 2005) encounters a similar situation; he presents a view on message passing algorithms as a way to approximate BNs by distributed divergence minimisation. It is pointed out that "A large network can be divided into pieces, each of which is approximated variationally, yielding an overall variational approximation to the whole network.". Every factor of the distribution factorisation gets approximated by minimising a local version of divergence and a message passing scheme between the factors distributes the approximation to other factors. This is repeated until some convergence criterion is reached. It is stated that optimising the local error is generally not the same as optimising the global error, but that results are similar.

The difference to our setup is that we do not choose to optimise a divergence, but try to find an approximation such that the divergence is acceptably small. Instead of a specific class of approximation families (such as Gaussians), we work with sets of conditional probabilities with reduced conditioning variables instead. Also, we aim to run the local approximation only once and not repeatedly.

Remark 2.34. *We give two important comments here before looking at some examples.*

(i) Obviously, if the only simplifications are made for nodes with no children, then there is no sequential error effect.

(ii) One can use the algorithm to determine which edge signifies the 'weakest conditional dependence' by tracking which of the edges with loss greater zero would be removed first.

2.4 Numerical experiments

In this section we examine two types of data sets: one smaller and artificial, the other large and taken from a real-world case study. We used the `gRain` R-package in order to sequentially build a BN that, in step i, contains variables X_1 to X_{i-1} and initially adds full information about X_i from the ET data. Then, the `gRain` package gave us an efficient way of querying for conditional and joint distributions whenever we tested different conditional independencies. Once the network has been built, different inferences can be done and probabilities queried for with the help of `gRain` too.

2.4.1 Application to a toy example: artificial data set

The first data set we examine was created to resemble the real-world data we are concerned with later. Most of the data sets we have seen have similar variable structures which roughly are as

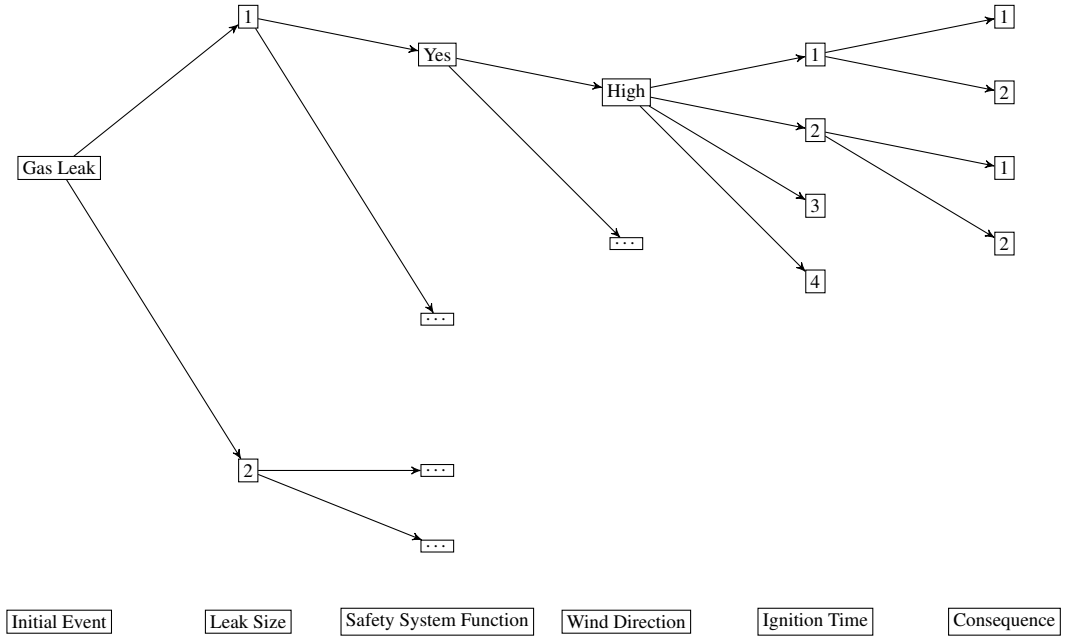

Figure 2-4: *Upper part of the artificial data ET.*

follows. The first variables in the tree usually describe conditions of the environment of an accident or functionality of safety systems. Following this setup, there is information about potential ignition times. Based on all these specifications we experience different consequences such as explosions, flash fires and the like.

We created variables called `Leak Size`, `Safety System Function`, `Weather`, `Ignition Time`, `Consequence`. The number of states for `Ignition Time` is four, all other variables are binary. We display an upper branch part of the tree in Figure 2-4. The chosen probabilities can be found in the appendix as Table A-1 which displays the full ET in table format. Each row corresponds to one full path in the tree.

To demonstrate the sensitivity of the number of obtained network edges to changes in the selected threshold, α was varied in the interval $[0, 0.001]$, such that $\alpha \in \{0 + k \cdot 0.00001\}$. (Due to numerical imprecisions, we actually start with $\alpha = 10^{-10}$ instead of zero.) Referring back to Subsection 2.3.2, this corresponds to a maximal $m \approx 0.022$ or, for example, a probability of exceeding a maximum difference of 0.1 bounded by 0.05.

We present the network structure for the selected thresholds $\alpha \in \{10^{-10}, 0.0005\}$ in Figure 2-5. The number of present edges for each α is shown in Figure 2-6. In the same figure there is shown a kind of root mean square error (RMSE) with the aim to gauge the 'global' error we make in our approximation. This RMSE was inspired by (Thornton, 2005) and computed in the following way:

For a fixed α, we obtain a network BN_α with CPTs $q(X_i|\mathrm{pa}(X_i))$ which we compare to $p(X_i|\mathrm{pa}(X_i))$ corresponding to the CPTs from the network obtained with threshold zero, BN_0.

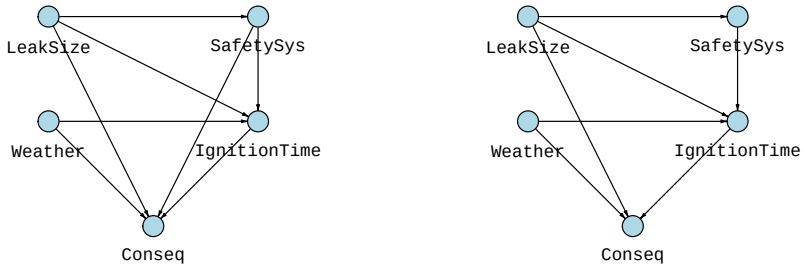

Figure 2-5: *Artificial data example: Network structure for the thresholds* $\alpha = 10^{-10}$ *and* $\alpha = 0.0005$.

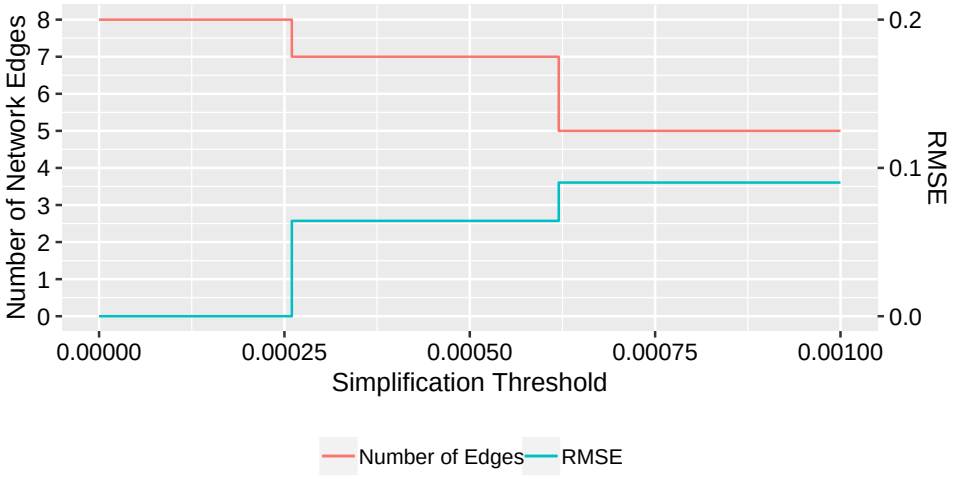

Figure 2-6: *Artificial data example: Number of edges and RMSE against threshold* α.

(Hence BN_0 is the network that only removes true conditional independencies.) We define

$$\mathrm{RMSE}_{\mathrm{BN}_0}(\mathrm{BN}_\alpha) := \sqrt{\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{a_i}\sum_{k=1}^{|\mathscr{X}_i|}|p(X_i = x_{ik}|\mathrm{inst}_i(j)) - q(X_i = x_{ik}|\mathrm{inst}_i(j))|^2}, \tag{2.14}$$

where $x_{ik} \in \mathscr{X}_i$ is the k-th possible state of X_i and $\mathrm{inst}_i(j)$ is the j-th instantiation of all a_i possible ones of the parent variables $\mathrm{pa}_p(X_i)$ of X_i in model BN_0 and $m = \sum_{i=1}^{n}\sum_{j=1}^{a_i}|\mathscr{X}_i|$ is the total number of compared probabilities. If X_i has no parents, then we just compare the marginal probabilities.

It was pointed out in Subsection 2.3.3 that the way we sequentially simplify parent sets causes that the Kullback-Leibler divergence only approximately decomposes along the simplified network. We compare the KLD $D(p(x_1,\ldots,x_n)\|q(x_1,\ldots,x_n))$ to the sum of expected KLDs $\mathbb{E}_{q_{(x_1,\ldots,x_i)}} D\left[p\left(x_{i+1}|x_i,\ldots,x_1\right)\|q\left(x_{i+1}|\mathrm{pa}\left(x_{i+1}\right)\right)\right]$ (with respect to q) from removing individual edges. This could be seen as comparing the 'global' error to the 'sum of local errors'; both quantities can be found in Figure 2-7 . Notice that the sum of local errors underestimates the global error as soon as there are two or more edges removed. However, it is generally also possible that the sum of local errors overestimates the global error, which could be a beneficial situation. We generated the

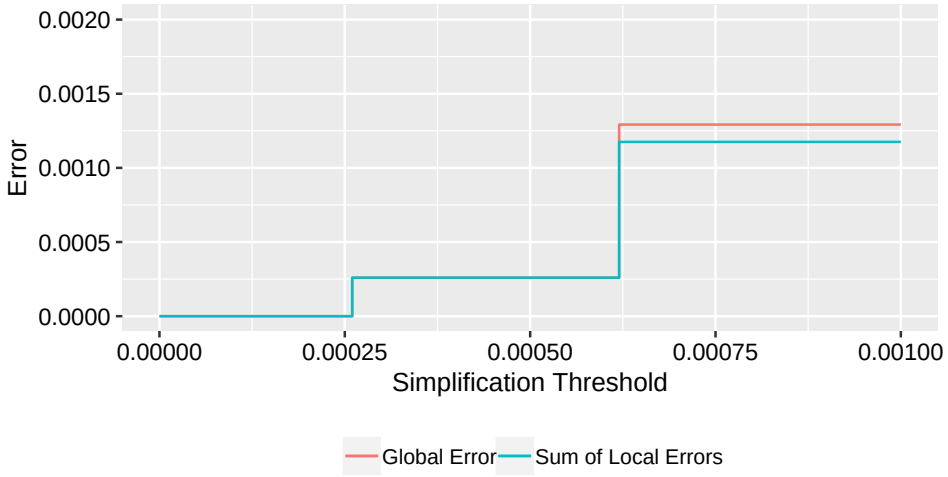

Figure 2-7: *Artificial data example: Comparison of the error structure against threshold* α.

following example where this is the case.

Example 2.35. *Consider an ET which includes the four binary variables* $\{A,B,C,D\}$ *described by Table A-2 in the appendix. Suppose we selected the threshold* $\alpha = 0.001$. *The approximation scheme firstly removes the edge* (A,B) *and thereafter the edge* (B,C). *We obtain a global error of approximately (rounded up)* 0.000855, *whereas the sum of local errors are approximately (rounded down)* $0.000199 + 0.000697 = 0.000896$. *The sum of the errors for individual removals thus overestimates the actual total error (in terms of the Kullback-Leibler divergence).*

In applications there could arise the following special case. The change of posterior probabilities in a node that we consider a prediction node (say, a node with no children, such as `Consequence`) may be more relevant compared to the change in posterior probabilities of 'interior' network nodes (certain nodes with both parents and children). Given any instantiation for the nodes without parents which are assumed to be typical input nodes of a model, simplifications of the network 'interior' might be rendered irrelevant. We can demonstrate this with the data set from the current section. Let `LeakSize`, `Weather` be the only observed 'input' nodes. Then the conditional probability function of $p(\mathtt{C}|\mathtt{L},\mathtt{W})$ in the original model with $\alpha \approx 0$ and the model for $\alpha = 0.0005$ are reported in Table 2.1. The edge between `SafetySys` and `Conseq` was eliminated, yet the specific network structure leads to no changes in the examined posterior probabilities, if no observation of `SafetySys` are entered.

2.4.2 Real-world application: Offshore event tree

Let us now turn to a data set that was obtained from a real-world study and supplied by DNV GL. It was produced with the Safeti Offshore software and consists of a data table with 24 (relevant)

L	W	$\alpha = 1 \cdot 10^{-10}$		$\alpha = 0.0005$	
		$p(\mathtt{C}=1\|\mathtt{L},\mathtt{W})$	$p(\mathtt{C}=2\|\mathtt{L},\mathtt{W})$	$p(\mathtt{C}=1\|\mathtt{L},\mathtt{W})$	$p(\mathtt{C}=2\|\mathtt{L},\mathtt{W})$
1	1	0.99241800	0.00758200	0.99241800	0.00758200
1	0	0.97657050	0.02342950	0.97657050	0.02342950
2	1	0.97888900	0.02111100	0.97888900	0.02111100
2	0	0.95811900	0.04188100	0.95811900	0.04188100

Table 2.1: *Artificial data example with two different thresholds: Posterior probabilities for the* `Consequence` *variable, given states for* `LeakSize` *and* `Weather`.

columns and 21644 rows. The corresponding full ET would contain more paths, but many paths that contained a zero probability edge were omitted. We have the following variables (with number of possible states in parentheses): `Name` (3), `bAD` (2), `bI` (2), `bB` (2), `bIC` (2), `iWea` (2), `iDir` (8), `iTime` (7), `iExp` (5), `iFireWater` (2), `iEarly` (2), `iHVAC` (2). There are other columns in the table that contain extra information, such as the ET path index of a row. To give a better context we describe the variables a bit more. `Name` simply denotes the possible types of initial event, in our case this is a 'small', 'medium' or 'large' leak in a certain area. The following variables `bAD`, `bI`, `bB`, `bIC` describe the functioning (yes / no) of various safety systems. `iWea` describes some weather conditions, usually given by temperature and `iDir` captures the wind direction on a wind rose; together these conditions can determine e.g. the movement of a gas cloud. `iTime` is a discretised ignition time variable that gives certain time intervals for which ignition likelihoods are being considered. The variables `iExp`, `iFireWater`, `iEarly`, `iHVAC` are all related to outcomes of the accident, for example, `iExp` encodes different types of explosions.

For any branch, if the conditional probabilities over all variable states do not sum to one (which happens sometimes due to rounding etc.) we scale them uniformly to do so. For example, for a binary conditional distribution on $\{y_1, y_2\}$, suppose $p(y_1|\mathbf{x}) + p(y_2|\mathbf{x}) \neq 1$. Then we rescale $p(y_1|\mathbf{x}) \leftarrow \frac{p(y_1|\mathbf{x})}{p(y_1|\mathbf{x})+p(y_2|\mathbf{x})}$ and similarly for $p(y_2|\mathbf{x})$.

The `iTime` variable is a time-to-event variable. In this case it encodes the probability of ignition in a certain time interval. (Only one ignition is possible.) The time intervals given correspond to 'immediate ignition' (ignition at $t = 0$), ignition in time interval $I_i = (t_{i-1}, t_i]$ and no ignition in the time frame considered. There are two probability columns associated with this variable. One describes the survival probability, i.e. probability that ignition occurs after time interval i and one column describing the conditional probability of ignition in time interval i, given no previous ignition. From this information we calculated the conditional distribution of ignition in a time interval, given the previous variables. (See Chapter 3 for details).

Figure 2-8 shows the obtained BN structure for running the algorithm with a threshold close to zero $\left(\alpha = 10^{-15}\right)$ to account for numerical imprecisions. (We note that `iHVAC` is an isolated note, since it was set to a fixed certain state in the data.)

The given ET data set contains some conditional distributions that are extremely similar and could, in principle, only differ because of the computational way they were assigned. For this real-world example, the probabilities were obtained from a physical model and so rounding effects

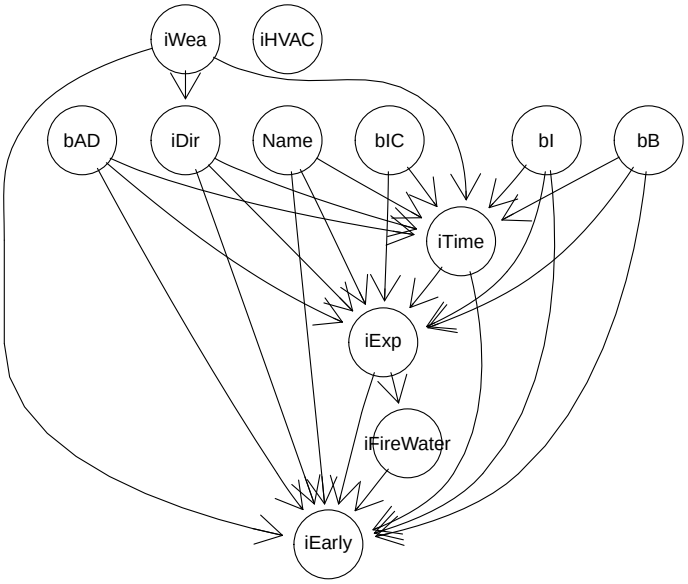

Figure 2-8: *Illustration of the obtained real-world network using* $\alpha = 1 \cdot 10^{-15}$.

could have created small differences. As an example, there is an edge between `iWea` and `iDir`. The probabilities found in the ET for `iDir` given `iWea` and irrespectively of the other variables' states look like in the following Table 2.2. This is before any of our own calculations have been done.

`iWea`	1	2
$p(\texttt{iDir} = 1 \mid \texttt{iWea})$	0.12082496750	0.12082503528
$p(\texttt{iDir} = 2 \mid \texttt{iWea})$	0.05797492084	0.05797508593
$p(\texttt{iDir} = 3 \mid \texttt{iWea})$	0.07812496698	0.07812503585
$p(\texttt{iDir} = 4 \mid \texttt{iWea})$	0.10935009401	0.10934989794
$p(\texttt{iDir} = 5 \mid \texttt{iWea})$	0.14230003502	0.14229996198
$p(\texttt{iDir} = 6 \mid \texttt{iWea})$	0.17902504784	0.17902494806
$p(\texttt{iDir} = 7 \mid \texttt{iWea})$	0.16314995834	0.16315004523
$p(\texttt{iDir} = 8 \mid \texttt{iWea})$	0.14925000946	0.14924998973

Table 2.2: *Two similar conditional distributions as found in the original ET data.*

The DAG for $\alpha = 10^{-12}$ is displayed in Figure 2-9 and shows an interesting effect. The threshold was increased and yet is still quite small, but the edges from `iWea` to `iDir` and `iFireWater` to `iEarly` have been removed. It could be the case that some of the dependencies the ET data contained was artificial since differences are of the order 10^{-5}. Very small thresholds could be used to detect such artificial dependencies originating from rounding effects.

Additionally, there was one new edge from `bIC` to `iEarly` created. The reason for this is

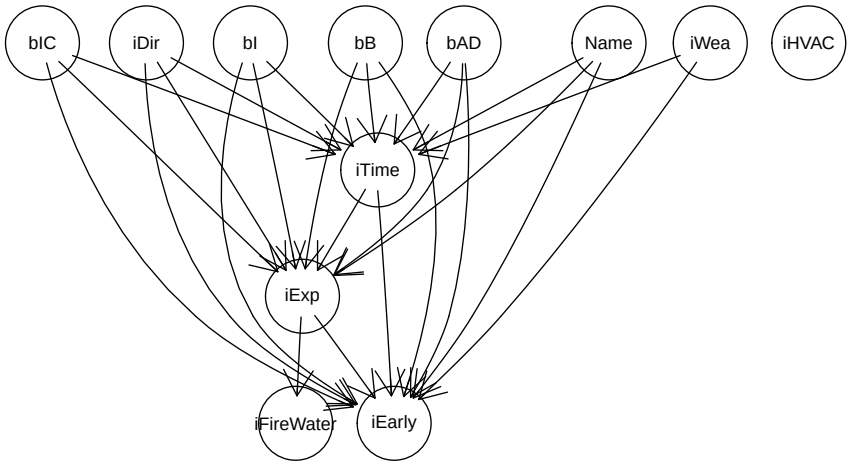

Figure 2-9: *Illustration of the obtained real-world network using* $\alpha = 1 \cdot 10^{-12}$.

that when the parent set for `iEarly` was determined, any removed edges from earlier steps could have affected the joint distribution involving `iEarly`. That is, the algorithm could create a non-monotonic behaviour as the threshold increases, but of course, for every simplification each parent set for X_i will always be a subset of $\{X_1, X_2, \ldots, X_{i-1}\}$.

This problem can be circumvented. One can use two stages, one of which serves as translation that removes true conditional independencies, the other purely for simplifications of the translated network. With a translation using $\alpha = 0$ first, all true conditional independencies are removed. Then in a second step that serves for simplification only, one could apply the algorithm with any kind of $\alpha > 0$ on the translated BN from the first step (such as in the next section) and never create additional parents. In this second step all initial sets of potential parents for a variable X_i are restricted to variables that are established to not be fully irrelevant to X_i, i.e. $\mathrm{pa}_{\mathrm{pot}}(X_i) \subseteq \{X_j : j \leq i-1, X_i \not\perp\!\!\!\perp X_j | \{X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_{i-1}\}\}$. (This also means that the second step runs much faster as fewer possibilities have to be considered so that there would be no bigger loss in runtime performance.)

We try this method on the same ET data and display the result in Figure 2-10. (Where $\alpha = 0$ was approximated by $\alpha = 10^{-15}$ in the first step.) It can be seen that the same edges (from `iWea` to `iDir` and from `iFireWater` to `iEarly`) were removed, however, the edge from `bIC` to `iEarly` has not been artificially created. After the first translation with $\alpha \approx 0$, the variables rendered irrelevant were not considered again as potential parents when the simplification step is running.

We compute the number of obtained network edges against different thresholds using the described two-step procedure; since this real-world data is numerically more sensitive we want to avoid these artificial parents due to approximation and numerical precision. The results can be seen in Figure 2-11. Notice that we have chosen a different granularity of the tested thresholds to account for the higher sensitivity in this data. For the first step we chose $\alpha = 1 \cdot 10^{-15}$ and in a

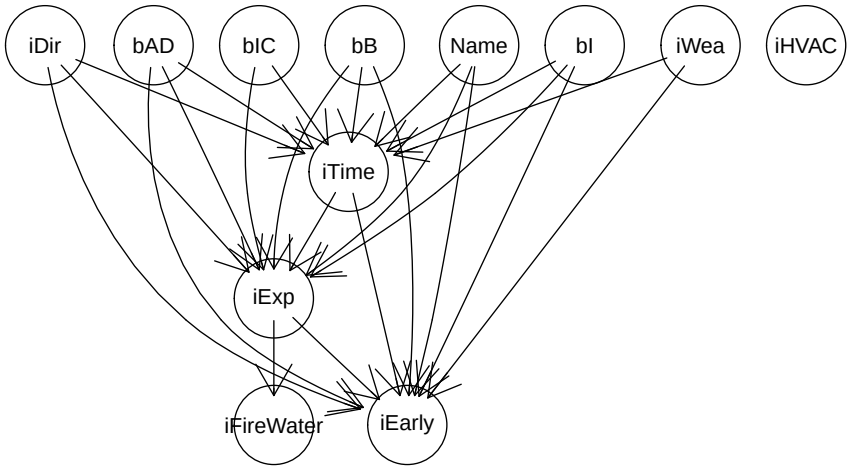

Figure 2-10: *Illustration of the obtained real-world network using* $\alpha = 1 \cdot 10^{-12}$ *and a two-step procedure.*

second step we varied $\alpha \in \{10^{-k} : k = -15 + 0.1 \cdot j, j = 0, \ldots, 110\}$. For the first run in the second step another edge gets removed and thereafter it is seen that three more edges can be removed at relatively small thresholds ($\alpha \leq 10^{-8}$).

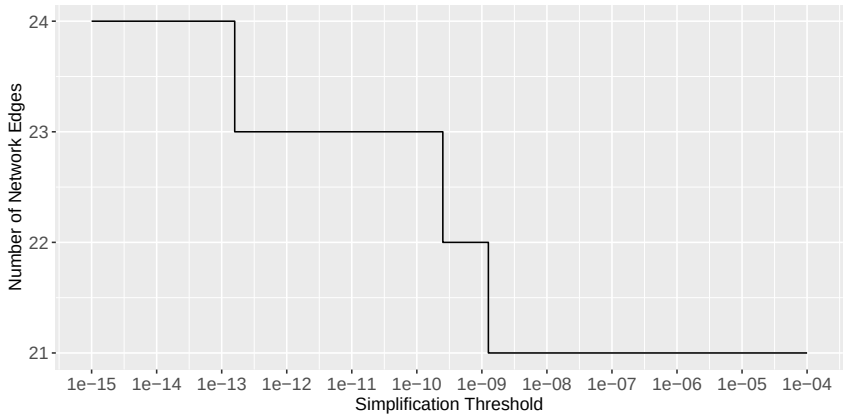

Figure 2-11: *The number of edges in the offshore network against different threshold values, second step.*

A big advantage of using this translation / simplification is clearly the easier visualisation of the variables and their dependencies by using a BN. While these large event trees are created automatically, this also means that often clear or purposefully created conditional independencies are hidden in a non-transparent table structure. In this specific example of Subsection 2.4.2 we have experienced that the data contained at least one 'almost' conditional independence, see Table 2.2. The translation using information measures allows for detecting such cases which might stem from inaccuracies of numerical simulations that generated the data. Furthermore, a practitioner could gauge both how strong links in such a network are by observing the thresholds at which

edges disappear or check which links are weakest / strongest by observing the order of edge removals.

2.5 The application of the simplification algorithm to Bayesian networks

We demonstrated with Algorithm 1 how ETs can essentially be regarded as BNs representing the same joint probability distribution. This means that the suggested simplifications equally can be applied when the algorithm starts with a Bayesian network structure. In this section we examine some BNs that appeared in earlier research.

One of the earliest articles on structural simplifications of BNs seems to be (Kjærulff, 1994), where edges are removed from the so-called moralised independence graph (we do not introduce this concept here). It is stated that, using message-passing for inference, the computational cost depends roughly on the size of the largest clique of this graph. For undirected graphs a clique is a subset of vertices such that every pair from those vertices are connected by an edge. Hence the aim to reduce the size of large clique(s), which are the main problem for inference in large BNs. The chosen approach to simplification consists of evaluating the deviation between correct clique potentials (essentially non-negative functions on the state space of the clique variables from which probability distributions can be calculated) and approximating ones by employing the CMI. Experiments are carried out on different networks.

Our method is closest to (Van Engelen, 1997). A number of edges is removed from a BN simultaneously but such that the set of removed edges does not contain two arcs with a common child. This constraint was imposed so that "the approximation scheme can select a near-optimal set of arcs for removal based on individual arc considerations alone with respect to the approximation error introduced and the reduced network complexity". Comparing to our algorithm this means that each parent set can be reduced by a maximum number of one parent. The approximation procedure (Van Engelen, 1997) computes the effect of removal for each edge individually using the CMI and combines this to the effect of removing several arcs by using an arc-divergence measure. The author also suggests a heuristic 'arc-connectivity measure' in order to find a near-optimal set of removal arcs and combines the divergence and connectivity measure to offer a trade-off between information loss and structure simplicity.

(Choi et al., 2012) uses a more direct approach to reduce network edges by considering evidence (i.e. instantiations of certain variables to certain states). The idea is that if some given evidence fixes the value of a variable Y, then links from Y to its children can be deleted. Now one might chose to remove the link between Y and its children $\mathrm{ch}(Y)$ also in case the evidence 'almost' fixes the value of Y in the sense that the conditional (posterior) distribution of Y given the evidence will not be a unit mass, but maybe spread between two values one of which has mass close to one. A drawback of this method is that we usually need to know the posterior to make a decision, but the authors give a work-around by iteratively using approximating networks and

updating posteriors until convergence.

The networks we examine include the famous Asia network, as appeared in (Lauritzen and Spiegelhalter, 1988); one version of the different Hepar networks developments, based on the work (Oniśko, 2003) and the Pathfinder network from (Heckerman et al., 1992). These choices are motivated by the different number of nodes and parameters of the networks. The data was downloaded from (bnlearn.com/bnrepository; accessed 15 July 2019) and from the same source we cite the network statistics displayed in Table 2.3.

Network Name	Asia	Hepar	Pathfinder
Number of Nodes	8	70	109
Number of Arcs	8	123	195
Number of Parameters	18	1453	72079
Average Markov Blanket Size	2.5	4.51	3.82
Average Degree	2	3.51	3.58
Maximum Size of Parent Sets	2	6	5

Table 2.3: *Network statistics for the Asia, Hepar, Pathfinder networks.*

Remark 2.36. *The degree of a node means the number of parents and children combined.*

We test the simplification capabilities of the algorithm using the three networks and thresholds α ranging in the interval $[0, 0.002]$, such that $\alpha \in \{0 + k \cdot 0.00001 | k = 0, 1, \ldots, 200\}$. (In the context of the inequalities from Subsection 2.3.2, the maximum value of 0.002 would correspond to $m(Y, \mathbf{X}, \mathbf{X_s}) \approx 0.032$ or, for example, a probability of exceeding a maximum difference of 0.1 bounded by 0.1. Hence, 0.002 can be regarded as a fairly large threshold for practical purposes.)

Figure 2-12 displays the Asia network structure for two ranges of thresholds.

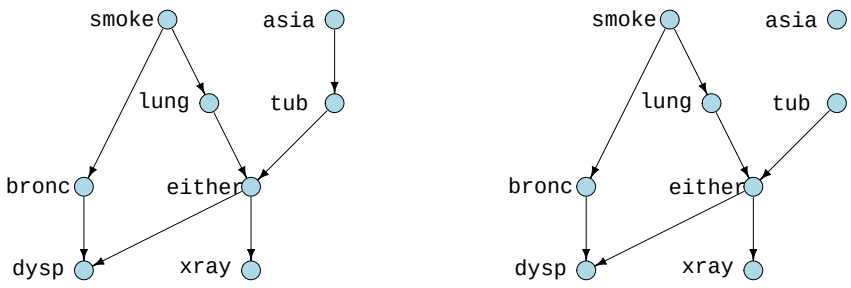

Figure 2-12: *Illustration of the Asia network structures for threshold ranges* $\alpha \in [0, 0.0004]$ *(left) and* $\alpha \in [0.0005, 0.0224]$ *(right).*

Generally, the structure for the Asia network does not change by many edges across the tested thresholds. Most of the encoded conditional dependencies are strong in an information theoretic sense such that for any variable removing one of its parents has large impact on the CPTs. This is supported by Figure 2-13 which shows the number of network edges as a function of the chosen

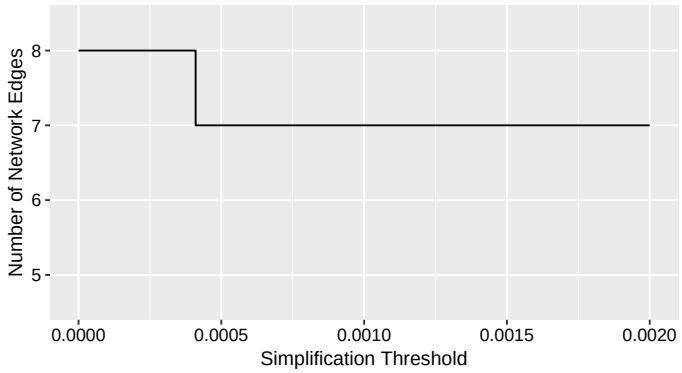

Figure 2-13: *The number of edges in the Asia network against different threshold values.*

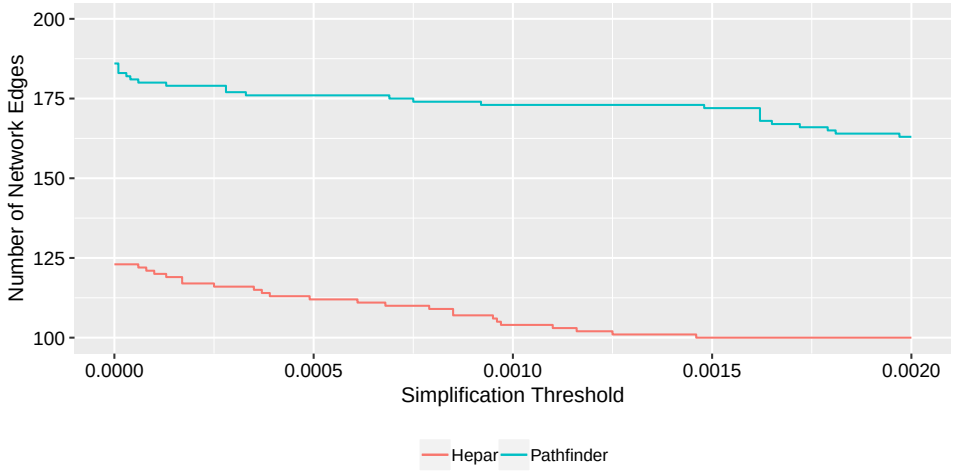

Figure 2-14: *Illustration of the number of network edges against different thresholds for the Hepar and Pathfinder networks.*

threshold. The first edge is removed using a quite small threshold value $\alpha < 0.0005$, whereas the next edge is removed only using a threshold in the neighbourhood of 0.02.

The Hepar and Pathfinder networks seem to have a number of 'weaker links' as is indicated by the steadily decreasing function in Figure 2-14. If we increase the threshold continuously, both of these nets loose edges quickly until there are longer and longer periods of no removals. Roughly between 9-10% of edges will be removed when using the maximal tested threshold.

In order to compare computation times, we used the R-package `microbenchmark`. We report the median running times of executing the algorithm 10 times for each threshold $\alpha \in \{0 + k \cdot 0.0001 | k = 0, 1, \ldots, 20\}$ in Figure 2-15. The stated times should only be compared in a relative way against each other for different thresholds and not be seen as an absolute performance indication as certainly both hardware and testing setup could be improved.

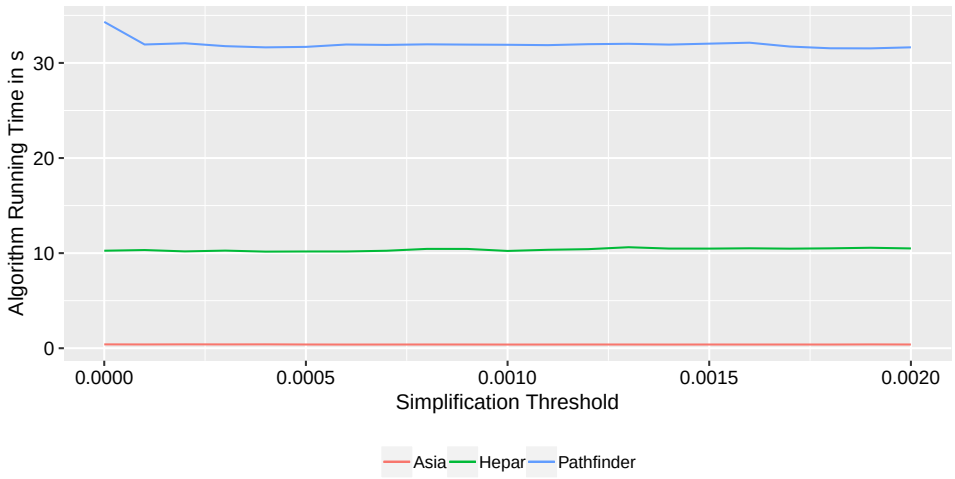

Figure 2-15: *Median algorithm running time against different threshold values.*

It is noteworthy that the computation times for the networks do not seem to depend on the chosen threshold (except for an initial drop in duration for the Pathfinder network). This could be due to the fact that, for some thresholds, essential parents are detected at once and the selection loop breaks, whereas for others, there need to be several iterations to conclude a parent set for a node.

Finally, the reader can see in Table 2.4 the initial network parameters (in parentheses) against the resulting parameters when using $\alpha = 0.0005$.

Network Name	Asia	Hepar	Pathfinder
Number of Nodes	8 (8)	70 (70)	109 (109)
Number of Arcs	7 (8)	112 (123)	176 (195)
Number of Parameters	17 (18)	1388 (1453)	57363 (72079)
Average Markov Blanket Size	2.25 (2.5)	4.2 (4.51)	3.45 (3.82)
Average Degree	1.75 (2)	3.2 (3.51)	3.23 (3.58)
Maximum Size of Parent Sets	2 (2)	6 (6)	4 (5)

Table 2.4: *Comparative network statistics for the Asia, Hepar, Pathfinder networks using simplification threshold* $\alpha = 0.0005$.

We conclude this part with some figures of the Hepar network for $\alpha = 0$ and $\alpha = 0.0005$. The next chapter develops some model extensions that are motivated by the safety risk background and can be included in the automated translation / simplification.

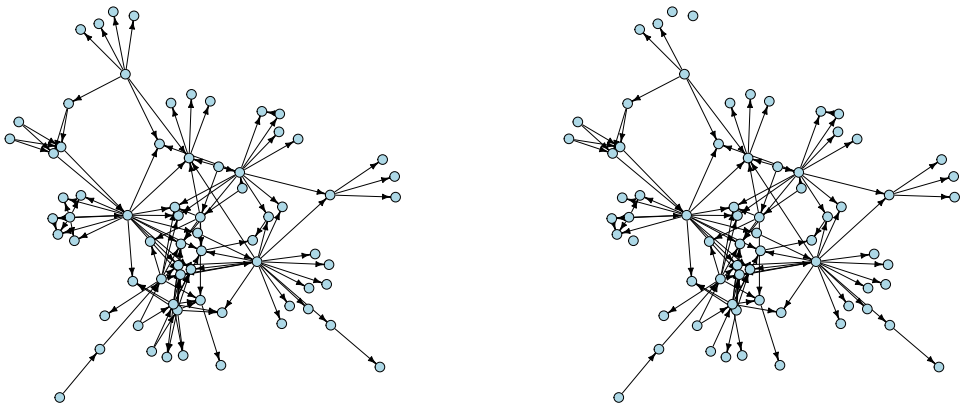

Figure 2-16: *The Hepar network for* $\alpha = 0$ *(left) and for* $\alpha = 0.0005$ *(right).*

CHAPTER 3

A MODEL EXTENSION TO SIMPLE HYBRID BAYESIAN NETWORKS FOR CONTINUOUS TIME

In the last chapter we showed how event trees can be translated automatically into corresponding discrete Bayesian networks using an information measure. We did not consider the nature of the random variables in the event tree, some of which are usually discretised versions of random variables that have a continuum of possible values or are conditioned on a continuum of values. We remarked earlier that this is one of the disadvantages of event trees; the inability to represent continuous objects.

In this chapter we consider some model extensions that focus on the idea of introducing continuous time into the network. We will use the discretised ET data for certain types of variables and construct during the translation continuous time versions which will enter the BN. This allows to compute more arbitrary probabilities, for example over time intervals that were not covered in any ET data. The construction of such variables relies on satisfying coherence criteria such that we recover the given ET probabilities even with the continuous variables.

A mixture of discrete and continuous nodes results in hybrid BNs. We will mention some facts about these kind of networks and then describe two simple extensions to our current translation model. The model extensions aim at the following:

i) To include a node with continuous values in an interval $(0, t_{max}]$ and which is characterised by a piece-wise constant (and truncated) hazard function and,

ii) to include a node with a finite number of states, but a continuum as underlying set.

Whenever a BN contains a mixture of types of random variables, we can speak of a hybrid Bayesian network (HBN). Mixed graphical models have been studied at least since (Lauritzen and Wermuth, 1989). Often the literature considers this special case of a mixture: the only continuous variables are of a Gaussian form, given their only discrete parents. Additionally it is usually assumed that the discrete variables do not have continuous parents. (Lauritzen, 1992) is one of the first articles that describes exact inference for such a special case.

In general however, computations for HBNs are much harder. (Salmerón et al., 2018) review inference modes for hybrid BNs. They describe the general inference problem as follows: Suppose that $\mathbf{X}$ is a set of both discrete and continuous variables and that $\mathbf{X_E} \subset \mathbf{X}$ is a set of observed random variables. We can consider the general inference problem to be the computation of $p(x_i|\mathbf{x_E})$ for each $X_i \in \mathbf{X_J} \subseteq \mathbf{X} \setminus \mathbf{X_E}$. If we let $\mathbf{X_D}$ and $\mathbf{X_C}$ denote the discrete and continuous variables in $\mathbf{X} \setminus \mathbf{X_E}$, the following equation is stated for the problem of computing a joint probability function which would be needed to obtain $p(x_i|\mathbf{x_E})$:

$$p(\mathbf{x_E}) = \sum_{\mathbf{x_D} \in \mathscr{X}_{\mathbf{x_D}}} \int_{\mathbf{x_C} \in \mathscr{X}_{\mathbf{x_C}}} p(\mathbf{x}, \mathbf{x_E}) d\mathbf{x_C}. \tag{3.1}$$

Inference hence quickly becomes difficult or impossible to do analytically, as the following example from (Langseth et al., 2009) shows.

Example 3.1. *Suppose T_1, T_2, T_3, T_4 are binary variables and let Z_1, Z_2 be two continuous variables such that $p(T_i = 1|z_1, z_2) = \frac{1}{1+e^{-(w_{i,1}z_1 + w_{i,2}z_2 + b_i)}}$, for constants w_i, b_i and $i = 1,2,3,4$ and $Z_j \sim \mathscr{N}(\mu_j, \sigma_j)$, $j = 1,2$. Then it is pointed out that*

$$p(T_1 = 1, T_2 = 1, T_3 = 1, T_4 = 1) = \frac{1}{2\pi\sigma_1\sigma_2} \int_{\mathbb{R}^2} \frac{e^{\left(-\Sigma_{j=1}^{2} \frac{\left(z_j - \mu_j\right)^2}{2\sigma_j^2}\right)}}{\prod_{i=1}^{4} \left(1 + e^{-(w_{i,1}z_1 + w_{i,2}z_2 + b_i)}\right)} dz_1 dz_2,$$

for which there is generally no known analytic expression. Hence, numerical methods would be required to do inference.

In the two next sections, we will develop some extensions to the network model that allow to carry out the translation and simplification algorithm in a very similar manner.

3.1 A continuous time-to-event node in Bayesian networks exemplified by ignition time modelling

In the Piper Alpha accident in 1988, more than 160 oil platform workers died from an explosion and possible consequences such as oil fires. Details of this disaster can be found in (Kletz, 2001). This event made it clear that ignitions of gas clouds, etc., pose a big operational threat for the oil and gas extraction and that ignition modelling should be an important part of appropriate risk models.

3.1.1 Ignition time variables in event trees

Let us introduce the way in which the current version of DNV GL's Safeti software stores results about ignition times and the corresponding probabilities.

For each analysed (accident) scenario there is a time frame $[0, t_{max}]$ for which ignition is regarded possible, depending on specifications. This time frame is partitioned into a set of time in-

tervals $I_j := (t_{j-1}, t_j]$, $j = 1, \ldots, m$, i.e. it holds $0 = t_0 < t_1 < \ldots < t_m = t_{\max}$ and $(0, t_{max}] = \cup_{j=1}^{m} I_j$. Additionally, let $I_0 = \{0\}$ and $I_{m+1} = (t_m, +\infty)$. The partition is described by a set of points $T = \{t_0 = 0, t_1, \ldots, t_m = t_{max}\}$, the set of interval boundaries.

Information about ignition probabilities are stored in the following way. Let A_d be the discrete random variable with state space being the ordered set $\{I_j\}_{j=0}^{m+1}$, where the event $\{A_d = I_j\}$, $(j \neq m+1)$ is interpreted as ignition in time interval I_j and the event $\{A_d = I_{m+1}\}$ is thought of as no ignition. This assumption can be justified by the idea that after a certain time the conditions have changed in a manner that ignition is seen as practically impossible. (For example, wind conditions could have moved a gas cloud into an area with no ignition sources or dissipated it to an undangerous density.)

In a typical ET, we are then given probabilities of the type

$$p_{j|j-1} := \mathbb{P}\left(A_d = I_j | A_d > I_{j-1}\right),$$

for $j = 1, \ldots m$, $I_0 = 0$ and $\{A_d > I_{j-1}\} := \{A_d = I_k, k > j-1\}$. This is completed by the specification of $p_0 := \mathbb{P}\left(A_d = I_0\right)$. This list describes the conditional probability of ignition in a certain time interval, given no ignition in an earlier time interval. The motivation for storing the probabilities in this way is given by the structure the ignition times are represented in the ET. Figure 3-1 shows the corresponding branches.

A list of marginal probabilities $\{p_j\}_{j=1}^{m}$, where $p_j := \mathbb{P}(A_d = I_j)$, can be obtained by using the relation

$$p_j = p_{j|j-1} p_{|j-1},$$

with

$$p_{|j-1} = 1 - \sum_{k=0}^{j-1} p_k$$

being the probability of no ignition up to (including) time interval $j-1$. That is, $p_{|j-1}$ is the probability of 'survival' beyond interval $j-1$. The specification of the mass function is completed by setting the probability of no ignition (within the considered maximum time frame) $p_{m+1} = 1 - \sum_{j=0}^{m} p_j$.

Figure 3-1 draws attention to another problem when using ETs for visualisation purposes and to gain structural insights. Suppose we consider larger t_{max} and / or finer grained time interval sets T. The part of the ET representing the ignition time grows quickly in size, but does not visualise any additional useful information. Every branch can be considered an indicator variable of whether there was ignition within a certain time interval I_i or not.

We suggest to re-model the ignition time variable to remedy these problems: Instead of a discrete variable, where the conditional probability of ignition is a constant for each time interval, we rather assume that the ignition time is governed by a hazard function that is constant within each time interval. This leads to a variable with a continuum of possible (time) values. The advantages of this approach are mainly the following. Firstly, we will have the possibility for queries that

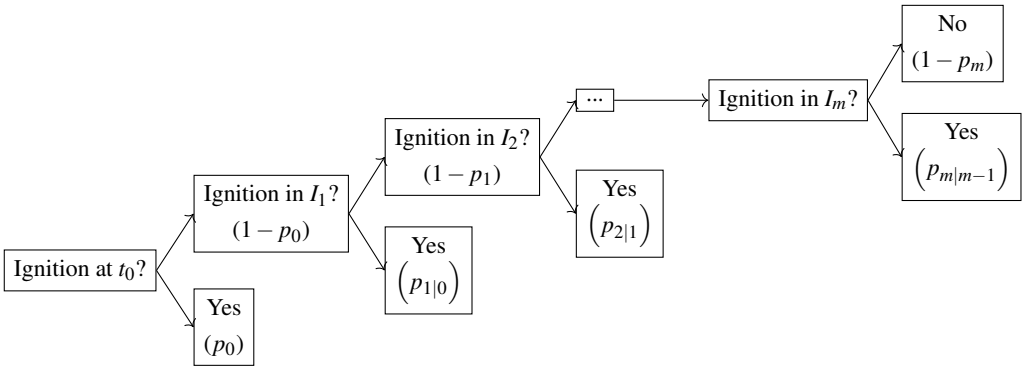

Figure 3-1: *Structure of the event tree part displaying ignition time results.*

involve time intervals which are not restricted by the break points of the time intervals. Next, we will have greater flexibility in the modelling choices, as one could generally choose different hazard functions. Thirdly, to work with this kind of continuous variable only requires storing the constant values of the hazard functions on the intervals, i.e. the same number of parameters as required by working with the CPT of the discrete version variable.

We will shortly review some necessary concepts from survival analysis before proceeding.

3.1.2 Some notions from survival analysis

There is a rich literature on survival analysis and we base our small review of definitions and ideas mainly on (Klein and Moeschberger, 2003) followed by some own calculations.

In risk and reliability analysis one quantity of interest is the time until a certain system component fails. Let us assume that a system component has a certain life time until failure represented by a non-negative random variable A or, more generally, that an event has a certain waiting time A to occurrence. If we know the cumulative distribution function (cdf) $F_A(t) := \mathbb{P}(A \leq t)$, $t \in \mathbb{R}$, of this random variable A and if its density $f_A(t)$ exists, then we can define the hazard function $h_A(t)$.

Definition 3.2. *(Hazard function)*
Let A be a non-negative, continuous random variable with cdf $F_A(t)$ and density $f_A(t)$. We define the hazard function $h_A : [0,\infty) \to [0,\infty)$ of A to be

$$h_A(t) := \frac{f_A(t)}{1 - F_A(t)},$$

whenever this expression exists.

Remark 3.3. *In the above definition we assumed that A is a continuous random variable for the ease of presentation. The concept hazard function can be defined in a more general setting where the distribution function of A has a finite number of discontinuities. For any continuous parts of such a more general function one can still use Definition 3.2. Discontinuities are treated using left-hand limits. For reference, see (Lawless, 2003).*

In the context of a model for the survival of a system component, the intuitive meaning of the hazard function is the conditional rate of failure 'around' time t.

One may see $f_A(t)dt$ as probability of failure in an infinitesimally small time interval dt: $\mathbb{P}(t \leq A < t+dt) = f_A(t)dt$. Then, considering that $1-F_A(t)$ is the probability of survival beyond time t, this leads to intuitive picture of $h_A(t)$ as the failure rate in an infinitely small time interval dt, conditional on the survival up to time t. It is sometimes useful to denote the function $1-F_A(t)$ by $S_A(t)$ and call it the survival function as it describes the probability of the event A not occurring before or at time t: $S_A(t) = \int_t^\infty f_A(\tilde{t})\,d\tilde{t}$. The next relation including continuous variables can be found in many textbooks; again we skip a proof.

Lemma 3.4. *The following equation relates the survival function to the hazard function:*

$$S_A(t) = \mathbb{P}(A > t) = \exp\left\{-\int_0^t h(\tilde{t})d\tilde{t}\right\}.$$

Next, we derive the probability of failure in a certain time interval $(t_i, t_{i+1}]$, given the survival up to time t_i.

$$\begin{aligned}\mathbb{P}(t_i < A \leq t_{i+1} | A > t_i) &= \frac{\mathbb{P}(\{t_i < A \leq t_{i+1}\} \cap \{A > t_i\})}{\mathbb{P}(\{A > t_i\})} = \frac{\mathbb{P}(t_i < A \leq t_{i+1})}{\mathbb{P}(A > t_i)} \\ &= \frac{F_A(t_{i+1}) - F_A(t_i)}{S_A(t_i)} \\ &= \frac{\left(1 - \exp\left\{-\int_0^{t_{i+1}} h(\tilde{t})d\tilde{t}\right\}\right) - \left(1 + \exp\left\{-\int_0^{t_i} h(\tilde{t})d\tilde{t}\right\}\right)}{\exp\left\{-\int_0^{t_i} h(\tilde{t})d\tilde{t}\right\}}.\end{aligned}$$

This immediately leads to the equality

$$\mathbb{P}(t_i < A \leq t_{i+1} | A > t_i) = 1 - \exp\left(-\int_{t_i}^{t_{i+1}} h(\tilde{t})d\tilde{t}\right). \tag{3.2}$$

The following examples will be useful for our applications in which we use piece-wise constant (and truncated) hazard functions.

Example 3.5. *(Constant hazard function)*
Suppose a hazard function is of the form $h(t) = \lambda \in \mathbb{R}_+$ for $t \geq 0$. Then $\mathbb{P}(A > t) = e^{-\int_0^t h(\tilde{t})d\tilde{t}} = e^{-\int_0^t \lambda d\tilde{t}} = e^{-\lambda t}$ and hence a constant hazard function implies a cdf F_A for A which is of the type of an exponential random variable.

Example 3.6. *(Piece-wise constant hazard function)*
Let $\{t_i\}_{i=0}^{m-1}$ be a set of jump points that define the intervals $(t_i, t_{i+1}]$ for $i = 0, 1, \ldots, m-1$ and let $t_0 = 0$. Furthermore, let $\mathbb{I}$ be the indicator function such that

$$\mathbb{I}\{t_i < t \leq t_{i+1}\} = \begin{cases} 1, & t_i < t \leq t_{i+1} \\ 0, & \text{else} \end{cases}$$

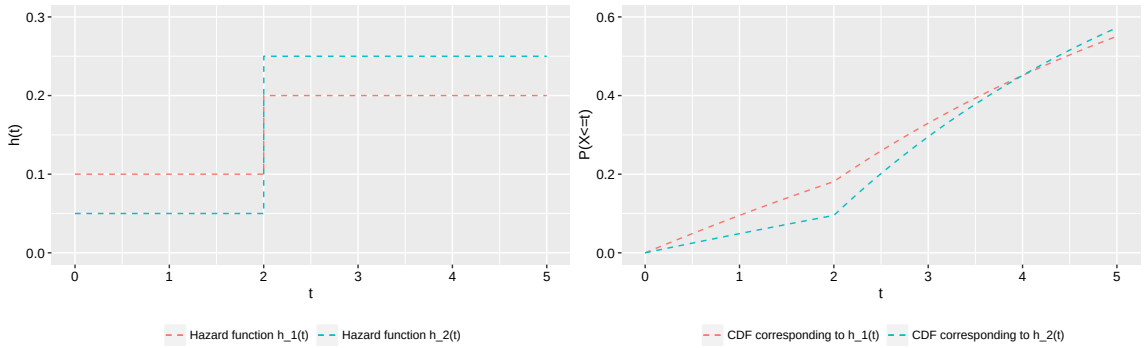

Figure 3-2: *Comparison of two piece-wise constant hazard functions $h_1(t)$, $h_2(t)$ and their corresponding cdfs.*

and

$$\mathbb{I}\{t_i < t < \infty\} = \begin{cases} 1, & t_i < t \\ 0, & \text{else.} \end{cases}$$

We can then define the piecewise constant hazard function $h(t)$ as

$$h(t) = \sum_{i=0}^{m-2} h_{i+1}\mathbb{I}\{t_i < t \leq t_{i+1}\} + h_m\mathbb{I}\{t_{m-1} < t < \infty\}$$

for non-negative constants $h_i, i = 1,\ldots,m$. We then observe that for any $t > 0$:

$$\begin{aligned}
\mathbb{P}(A > t) &= e^{-\int_0^t \sum_{i=0}^{m-2} h_{i+1}\mathbb{I}\{t_i<\tilde{t}\leq t_{i+1}\}+h_m\mathbb{I}\{t_{m-1}<\tilde{t}<\infty\}d\tilde{t}} \\
&= e^{-\int_0^t \sum_{i=0}^{m-2} h_{i+1}\mathbb{I}\{t_i<\tilde{t}\leq t_{i+1}\}d\tilde{t}} e^{-\int_0^t h_m\mathbb{I}\{t_{m-1}<\tilde{t}<\infty\}d\tilde{t}} \\
&= e^{-\sum_{i=0}^{m-2}\int_0^t h_{i+1}\mathbb{I}\{t_i<\tilde{t}\leq t_{i+1}\}d\tilde{t}} e^{-\int_0^t h_m\mathbb{I}\{t_{m-1}<\tilde{t}<\infty\}d\tilde{t}} \\
&= e^{-\sum_{i=0}^{m-2} h_{i+1}\int_{t_i}^{\max\{\min\{t,t_{i+1}\},t_i\}} d\tilde{t}} e^{-h_m\int_{t_{m-1}}^{\max\{t,t_{m-1}\}} d\tilde{t}} \\
&= e^{-\sum_{i=0}^{m-2} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+} e^{-h_m(t-t_{m-1})_+}
\end{aligned}$$

where we used the notation $(x)_+ := \max(x,0)$.
Illustrations of the specific choices of hazard functions
$h_1(t) = 0.1\mathbb{I}\{0 < t \leq 2\} + 0.2\mathbb{I}\{2 < t < \infty\}$ and $h_2(t) = 0.05\mathbb{I}\{0 < t \leq 2\} + 0.25\mathbb{I}\{2 < t < \infty\}$ together with the corresponding cdfs are shown in Figure 3-2 for the interval $0 \leq t \leq 5$.

Example 3.7. *(Piece-wise constant 'quasi' hazard function on a finite time horizon)*
In applications, hazards rates may be identified only for a finite time horizon such that the hazard function is truncated at the right. This can be compared to the situation of right-censored data which is not uncommon for lifetime data. In this case observations have a cut-off time $T_{\max}$ and if the event of interest has not been observed until $T_{\max}$, data is set to a certain state. Now suppose $\hat{h}(t) = \sum_{i=0}^{m-1} h_{i+1}\mathbb{I}\{t_i < t \leq t_{i+1}\}$ with $t_m < \infty$ is such a 'quasi' truncated hazard function. Then for the function given by $\hat{f}(t) = \hat{h}(t)\left(1-\hat{F}(t)\right)$ for all $t \geq 0$, we have $\int_{-\infty}^{\infty} \hat{f}(t)\,dt \neq 1$. This means $\hat{f}$

does not define a density. In fact, consider

$$\begin{aligned}
\int_{-\infty}^{\infty} \hat{h}(t)\left(1-\hat{F}(t)\right)dt &= \int_0^{\infty} \sum_{i=0}^{m-1} h_{i+1}\mathbb{I}\{t_i < t \leq t_{i+1}\} e^{-\sum_{i=0}^{m-1} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+} dt \\
&= \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} h_{i+1} e^{-\sum_{i=0}^{m-1} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+} dt \\
&= \sum_{i=0}^{m-1} h_{i+1} e^{-\sum_{j=1}^{i} h_j |I_j|} \left(\frac{1-e^{-h_{i+1}|I_{i+1}|}}{h_{i+1}}\right) \\
&= \sum_{i=0}^{m-1} \left(e^{-\sum_{j=1}^{i} h_j |I_j|} - e^{-\sum_{j=1}^{i+1} h_j |I_j|}\right) \\
&= 1 - e^{-\sum_{i=1}^{m} h_i |I_i|} = 1 - \hat{S}(t_m).
\end{aligned}$$

Hence we need to scale $\hat{f} = \hat{h}\left(1-\hat{F}\right)$ *by* $K = \frac{1}{1-e^{-\sum_{i=1}^{m} h_i |I_i|}}$ *to make it a proper density. In general, this corresponds to conditioning on the time-to-event being smaller than the censoring time, i.e. for our case conditioning on* $\{0 < A \leq t_m\}$*. We recalculate (a conditional)* $S(t)$ *and* $h(t)$*: For an arbitrary* t*, let* $t_k = \max_{t_j \in T}\{t_j \leq t\}$*, then*

$$\begin{aligned}
S(t) &= 1 - \int_0^t f(\tilde{t})\,d\tilde{t} = 1 - K\int_0^t \hat{f}(\tilde{t})\,d\tilde{t} \\
&= 1 - K\int_0^{t_k} \hat{f}(\tilde{t})\,d\tilde{t} - K\int_{t_k}^t \hat{f}(\tilde{t})\,d\tilde{t} \\
&= \frac{e^{-\sum_{i=0}^{m-1} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+} - e^{-\sum_{i=1}^{m} h_i |I_i|}}{1-e^{-\sum_{i=1}^{m} h_i |I_i|}},
\end{aligned}$$

$$\begin{aligned}
h(t) &= \frac{f(t)}{S(t)} = \frac{K\hat{h}(t)\left(1-\hat{F}(t)\right)}{S(t)} \\
&= \hat{h}(t)\,\frac{e^{-\sum_{i=0}^{m-1} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+}}{e^{-\sum_{i=0}^{m-1} h_{i+1}(\min\{t,t_{i+1}\}-t_i)_+} - e^{-\sum_{i=1}^{m} h_i |I_i|}} \\
&= \frac{\hat{h}(t)}{\left[1-e^{-\sum_{i=k+2}^{m} h_i |I_i| - h_{k+1}(t-t_k)}\right]}.
\end{aligned}$$

For future reference, we define the scaling factor for such a $\hat{h}$ *to be*

$$c(t) = \left[1-e^{-\sum_{i=k+2}^{m} h_i |I_i| - h_{k+1}(t-t_k)}\right]^{-1}.$$

So far we implicitly assumed that $\mathbb{P}(A=t)=0$ for all $t \geq 0$. In some cases though, we might want to account for the possibility of the event A happening (e.g. an item failing) instantly, say directly at time $t=0$ so that $\mathbb{P}(A=0)>0$. In this situation one can easily modify the variable A to become a mixed random variable consisting of two parts. The point mass at instant $t=0$ and

the continuous part described by a hazard function for $t > 0$[1]:

$$\begin{aligned}\mathbb{P}(A \leq t) &= \mathbb{P}(0 < A \leq t | A > 0)\,\mathbb{P}(A > 0) + \mathbb{P}(A = 0) \\ &= \left(1 - \exp\left(-\int_0^t h(\tilde{t})d\tilde{t}\right)\right)(1 - \mathbb{P}(A = 0)) + \mathbb{P}(A = 0)\,. \end{aligned} \tag{3.3}$$

In the next subsection we describe how these notions connect with the suggested model extension for ignition times.

3.1.3 The distribution of ignition times using piece-wise constant hazard functions

Let us remember that DNV GL's current ignition time model generates a set of conditional probabilities

$$p_{j|j-1} := \mathbb{P}(A_d = I_j | A_d > I_{j-1})\,,$$

together with $p_0 := \mathbb{P}(A_d = I_0)$. The probability of ignition within the considered finite time window is given by $p_0 + \sum_{j=1}^m p_j$, where p_0 can be seen as the probability of 'immediate ignition', $\sum_{j=1}^m p_j$ can be seen as the probability of 'delayed ignition' and $1 - (p_0 + \sum_{j=1}^m p_j)$ can be thought of as the probability of 'no ignition'.

This categorisation of probabilities has practical relevance. Immediate ignition concerns only the time point $t = 0$ and leaves no time for any kind of safety measures in practice. On the other hand, if there will be a delayed ignition, there is time for safety actions and so the exact time of ignition may be crucial. After a certain time has passed (the time frame of consideration), conditions change in such a manner that ignition at any further time point is considered impossible in principle, e.g. toxic clouds have moved away from ignition sources and evaporated. We will thus focus on translating the 'delayed ignition part' into continuous time.

The set $\left\{p_{j|j-1}\right\}_{j=1}^m$ can be used as a basis to generate an approximate continuous-time probability function with the help of a scaled, piece-wise constant hazard function. A hazard function that is constant over time intervals seems to be natural assumption for many applications and we outlined in Example 3.7 how to define such a function.

Remark 3.8. *Besides (piece-wise) constant hazard functions, other choices, such as the 'bathtub shape' are popular. These represent a common reliability idea, that items have a high chance of failing initially or towards the end of a specified time frame. This assumption is not characteristic for our background, as we are not typically dealing with 'lifetimes' of components, but for example with clouds of toxic chemicals and ignition sources or similar objects.*

For each time interval I_i, $i = 1, \ldots, m$, we are given exactly one conditional probability $p_{i|i-1}$ which will be used to calculate a hazard rate $h_i := \hat{h}(t)$ for $t_{i-1} < t \leq t_i$. After doing that for all

[1] In order to make the total probabilities sum to one, the hazard function will be such that the continuous part integrates to $\mathbb{P}(A \neq 0)$. For simplicity, $h(t)$ can be still thought of being defined for $t = 0$, as this would change it only on a zero set.

these time intervals, we obtain our (unscaled) quasi-hazard function of the form

$$\hat{h}(t) = \sum_{i=1}^{m} h_i \mathbb{I}\{t_{i-1} < t \leq t_i\}.$$

By (3.2), we choose to equate

$$p_{i|i-1} = 1 - \exp\left(-h_i |t_{i+1} - t_i|\right), \tag{3.4}$$

from which we find the values $\{h_i\}_{i=1}^m$ as

$$h_i = \frac{-\log\left(1 - p_{i|i-1}\right)}{|t_i - t_{i-1}|}. \tag{3.5}$$

The hazard rates determined via Equation 3.5 lead to a density-type function that will integrate to the total probability of 'delayed ignition', i.e. $\sum_{j=1}^m p_j$. To define a hazard function that leads to a proper density function, we use a scaled version, as in Example 3.7:

$$h(t) = c(t) \sum_{i=1}^{m} h_i \mathbb{I}\{t_{i-1} < t \leq t_i\}.$$

The supplied data sets included a discrete random variable A_d, as outlined. We transformed this variable to a partly continuous random variable A by approximating the 'delayed ignition' part of it by a piece-wise constant hazard function and keeping the 'immediate ignition' and 'no ignition' probabilities as they were.

The conditional cdf for $A|\{0 \leq A \leq t_m\}$ can then be determined by using (3.3) and Example 3.7 as

$$\mathbb{P}(A \leq t | 0 \leq A \leq t_m) = \left(1 - e^{-\int_0^t c(\tilde{t}) \sum_{i=1}^m h_i \mathbb{I}\{t_{i-1} < \tilde{t} \leq t_i\} d\tilde{t}}\right) \mathbb{P}(A > 0) + \mathbb{P}(A = 0).$$

3.1.4 Simplification aspects in the translation algorithm

In this subsection we show how to extend the translation algorithm to include a time-to-event variable of the same type such as the delayed ignition probabilities from the last subsection. We constructed a variable that can be expressed with the help of a piece-wise constant quasi hazard functions, but the same principle should be valid for any type of random variables for which the relevant quantities can be computed.

Consider the variable A with a continuum of states that is to be added to the network. We assume that there is a (possibly empty) set of discrete variables $\mathbf{X} = \{X_1, \ldots, X_d\}$ preceding A. For each configuration of values $\mathbf{x}$ of the $\mathbf{X}$, we furthermore assume that there exists a quasi hazard function $\hat{h}_{A|\mathbf{x}}(t)$ that depends on $\mathbf{x}$. Since we will only be working with one continuous variable here, we suppress A in the notation and so, for example, $\hat{h}_{\mathbf{x}}(t)$ shall mean $\hat{h}_{A|\mathbf{x}}(t)$ and so forth.

By earlier construction, the proper form of the hazard function $h_{\mathbf{x}}(t)$ is given by

$$h_{\mathbf{x}}(t) = c(t) \sum_{i=1}^{m} h_{\mathbf{x},i} \mathbb{I}\{t_{i-1} < t \leq t_i\}.$$

We can obtain a density function $f_{\mathbf{x}}$ for A by setting

$$f_{\mathbf{x}}(t) = h_{\mathbf{x}}(t)(1 - F_{\mathbf{x}}(t)), t \in \mathbb{R}_0^+, \tag{3.6}$$

where $1 - F_{\mathbf{x}} = S_{\mathbf{x}}$ is the 'survival function' for A given $\mathbf{x}$ and is expressed as

$$1 - F_{\mathbf{x}}(t) = \exp\left\{-\int_0^t h_{\mathbf{x}}(\tilde{t}) d\tilde{t}\right\}.$$

When A is added to the network, we determine the information loss when removing one of the potential (discrete) parent variables by now computing the relevant CMIs (the expected KLDs) using the conditional densities obtained for the continuous random variable conditioned on different subsets of the potential parents. Hence, it is again possible to check whether A is invariant in the outcomes of certain variables of $\mathbf{X}$, or if not, by how much it depends on these outcomes in an information-theoretic sense.

Since the following expressions are a bit more involved, we refrain from showing the derivation for the entropy and cross-entropy separately, but work with the KLD directly. The KLD can be expressed using the hazard rates directly. The following expression gives the KLD between two densities of the same type as in Equation 3.6. To be more precise, suppose $\mathbf{X}_a \subseteq \mathbf{X}$, $\mathbf{X}_b \subseteq \mathbf{X}$ are two sets of conditioning variables for A and $\mathbf{x}_1 \in \mathscr{X}_a$ and $\mathbf{x}_2 \in \mathscr{X}_b$ are state values for $\mathbf{X}_a$ and $\mathbf{X}_b$. Then

$$\begin{aligned} D(f_{\mathbf{x}_1} \| f_{\mathbf{x}_2}) &= \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt \\ &= \sum_{i=0}^{m-1} \Big[C_{1,i}[\phi_{\mathbf{x}_1}(t_i) - \phi_{\mathbf{x}_1}(t_{i+1})] + C_{2,i}[\phi_{\mathbf{x}_1}(t_i) - (1 + h_{\mathbf{x}_1,i+1}|I_{i+1}|)\phi_{\mathbf{x}_1}(t_{i+1})]\Big], \end{aligned}$$

where we set

$$K_1 = \frac{1}{1 - \phi_{\mathbf{x}_1}(t_m)}, \qquad K_2 = \frac{1}{1 - \phi_{\mathbf{x}_2}(t_m)}$$

and let

$$C_{1,i} = K_1 \left(\log\frac{K_1}{K_2} + \log\frac{h_{\mathbf{x}_1,i+1}}{h_{\mathbf{x}_2,i+1}} + \sum_{j=1}^{i} |I_j| (h_{\mathbf{x}_2,j} - h_{\mathbf{x}_1,j})\right)$$

and

$$C_{2,i} = \frac{K_1 (h_{\mathbf{x}_2,i+1} - h_{\mathbf{x}_1,i+1})}{h_{\mathbf{x}_1,i+1}}.$$

The details of the derivation of this expression can be found in Appendix A.5.

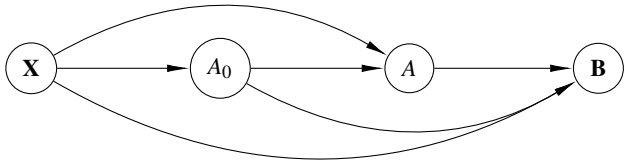

Figure 3-3: *Ignition time application: Continuous-time node A with pre-node* A_0.

Algorithm 4 describes the implementation of this new idea as part of the existing algorithm by editing the find-parents function.

Algorithm 4 Inclusion of a piece-wise constant hazard function type variable into the network.

Prerequisite: The network translation for $\mathbf{X}$ has been carried according to Algorithm 2.
Input: Distributions $p(A_d|\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}$, $p(X_j|\mathrm{pa}(X_j))$ for $j = 1,\ldots,d$.
Output: Parent set $\mathrm{pa}(A)$, such that $CMI(A, \{\mathbf{X} \setminus \mathrm{pa}(A)\}|\mathrm{pa}(A)) \leq \alpha$.

1. Assign an initial set of potential parents $\mathrm{pa}_{\mathrm{pot}}(A) := \{X_d,\ldots,X_1\}$.
2. Initialise a set of essential parents: $\mathrm{ess}(A) := \{\}$.
3. While $\left(\mathrm{pa}_{\mathrm{pot}}(A) \setminus \mathrm{ess}(A) \neq \emptyset\right)$ do:
   (a) For each $X_k \in \mathrm{pa}_{\mathrm{pot}}(A) \setminus \mathrm{ess}(A)$ do:
      i. If not available from the previous iteration, compute $p(A_d|\mathrm{pa}_{\mathrm{pot}}(A) \setminus X_k)$.
      ii. Compute the corresponding hazard function $h_{\tilde{\mathbf{x}}_k}$ from $p(A_d|\mathrm{pa}_{\mathrm{pot}}(A) \setminus X_k)$, according to Equation 3.5.
      iii. Compute $\alpha_k := CMI(A, \mathbf{X} \setminus \{\mathrm{pa}_{\mathrm{pot}}(A) \setminus X_k\}|\mathrm{pa}_{\mathrm{pot}}(A) \setminus X_k)$.
      iv. If $\alpha_k > \alpha : \mathrm{ess}(A) \leftarrow \mathrm{ess}(A) \cup X_k$.
   (b) If $(\{\alpha_k : \alpha_k \leq \alpha\} \neq \emptyset)$: Remove the $X_{\tilde{k}}$ for which $\tilde{k} = \mathrm{argmin}_k\{\alpha_k : \alpha_k \leq \alpha\}$ from $\mathrm{pa}_{\mathrm{pot}}(A)$:
   $$\mathrm{pa}_{\mathrm{pot}}(A) \leftarrow \mathrm{pa}_{\mathrm{pot}}(A) \setminus X_{\tilde{k}}.$$
   Else: Stop and go to step 4.
4. Set $\mathrm{pa}(A) \leftarrow \mathrm{ess}(A)$.

Remark 3.9. *The hazard function* $h_{\tilde{\mathbf{x}}_k}$ *for the subset* $\{\mathrm{pa}_{\mathrm{pot}}(A) \setminus X_k\} \subset \mathbf{X}$ *in step 3(a)(ii) was obtained by first computing the conditional probabilities using the discrete variable* A_d *in step 3(a)(i). Naturally, if one would start with a continuous time variable A initially, it is also possible to go a different way and obtain* $h_{\tilde{\mathbf{x}}_k}$ *directly from* $h_{\mathbf{x}}$. *We refrained from this as our aim here is only to re-model* A_d.

To work with the data sets in a way that distinguishes between 'immediate ignition', 'delayed ignition' and 'no ignition' conveniently, we decided to construct a helper variable within the network that precedes A and describes the mode of ignition we are dealing with, see also Figure 3-3. Let this node be called $A_0 \in \{0,1,2\}$. It is interpreted in the following way. A_0 can be seen as a random variable for which value 0 corresponds to immediate ignition (i.e. $A_d = 0$), value 1 cor-

responds to delayed ignition (i.e. $A \in (0, t_{max}]$) and value 2 corresponds to no ignition within the time horizon of interest (i.e. $A \notin [0, t_{max}]$) .

One can specify the distribution of A for each of the cases separately as follows.
Case 1 $(A_0 = 0)$:

$$\mathbb{P}(A \leq t | A_0 = 0, \mathbf{X} = \mathbf{x}) = \mathbb{I}\{t \geq 0\}$$

Case 2 $(A_0 = 1)$:
In this case we look at the truncated distribution

$$\begin{aligned} \mathbb{P}(A \leq t | A_0 = 1, \mathbf{X} = \mathbf{x}) &= \mathbb{P}(A \leq t | 0 < A \leq t_{max}, \mathbf{X} = \mathbf{x}) \\ &= \frac{\mathbb{P}(0 < A \leq \min\{t, t_{max}\} | \mathbf{X} = \mathbf{x})}{\mathbb{P}(0 < A \leq t_{max} | \mathbf{X} = \mathbf{x})} \end{aligned}$$

Case 3 $(A_0 = 2)$:

$$\mathbb{P}(A \leq t | A_0 = 2, \mathbf{X} = \mathbf{x}) = \begin{cases} 0 & , t \in [0, t_{\max}] \\ 1 & , t \geq \tilde{t} > t_{max} \end{cases},$$

for some arbitrary $\tilde{t} > t_{max}$.

In the process of including the variable A into the network we transformed conditional mass functions for $A_d | \mathbf{X} = \mathbf{x}$ to hazard functions $h_{\mathbf{x}}(t)$ based on Equation 3.5. We want to visualise the effect of this transformation for the comparison of two conditional mass functions / densities. Consider the real-life offshore QRA example we used in Section 2.5. The upper plot in Figure 3-4 displays two sets of conditional probabilities of 'delayed' ignition for `iTime`. In particular, the blue dots correspond to

$$\mathbb{P}(\texttt{iTime} = i | \texttt{iTime} > i-1, \texttt{iDir} = 1, \texttt{iWea} = 1, \texttt{bIC} = 1, \texttt{bB} = 1, \texttt{bI} = 1, \texttt{bAD} = 1, \texttt{EventName} = 1)$$

and the orange dots to

$$\mathbb{P}(\texttt{iTime} = i | \texttt{iTime} > i-1, \texttt{iDir} = 1, \texttt{iWea} = 1, \texttt{bIC} = 1, \texttt{bB} = 1, \texttt{bI} = 1, \texttt{bAD} = 1),$$

for $i = 1, \ldots, 5$. This is a typical case for part of the KLD computation whereby we compute the information divergence when approximating the 'blue dots' probabilities by the 'orange dots' probabilities. The lower plot of Figure 3-4 exhibits the (quasi-)hazard functions obtained by transforming the above conditional ignition probabilities according to Equation 3.5, but before normalising them. That is, the blue line shows $\hat{h}_{\mathbf{X}=\mathbf{x}}(t)$ where $\mathbf{X} = \mathbf{x}$ corresponds to $\texttt{iDir} = 1, \texttt{iWea} = 1, \texttt{bIC} = 1, \texttt{bB} = 1, \texttt{bI} = 1, \texttt{bAD} = 1, \texttt{EventName} = 1$ and the orange line shows $\hat{h}_{\mathbf{X}=\mathbf{x}}(t)$ where $\mathbf{X} = \mathbf{x}$ corresponds to $\texttt{iDir} = 1, \texttt{iWea} = 1, \texttt{bIC} = 1, \texttt{bB} = 1, \texttt{bI} = 1, \texttt{bAD} = 1$. For both functions $t \in (t_0, t_{\max}) = (0, 3600)$.

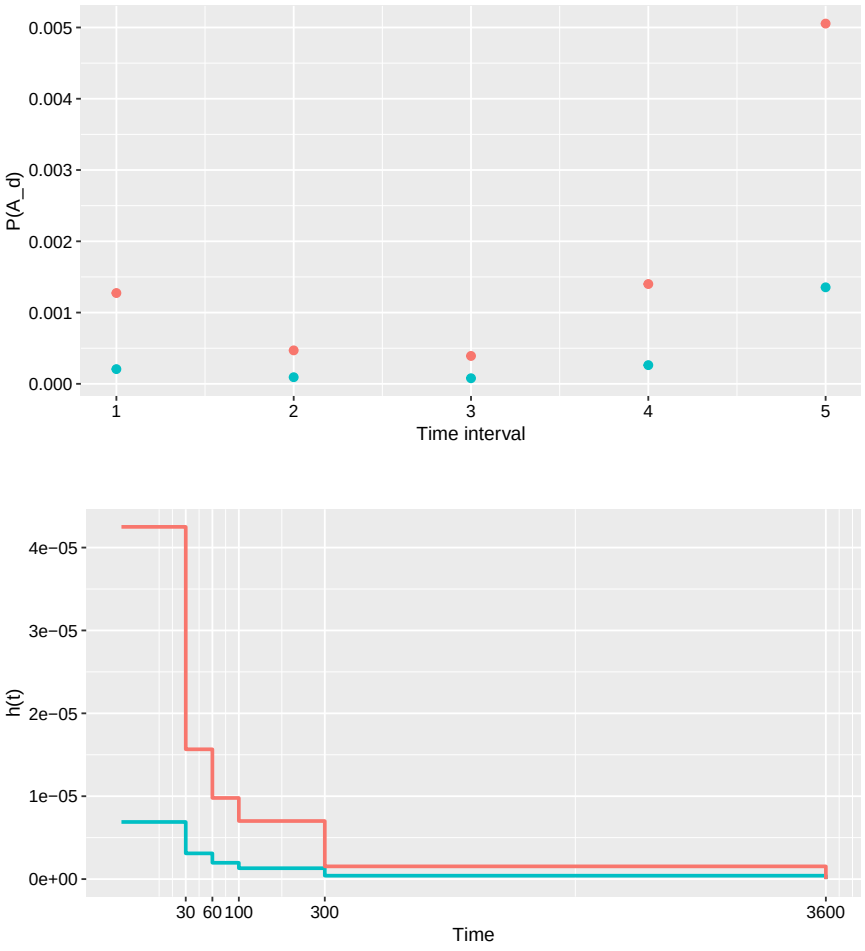

Figure 3-4: *Comparison of conditional probabilities of ignition (upper plot) and the corresponding (quasi) hazard functions (lower plot), given two different conditioning sets.*

3.2 A model for the inclusion of discrete nodes with continuous parents

In this section we show that with certain restrictions, discrete random variables with a continuum as underlying space can be introduced into our translation framework. This extension is motivated by considering possible children of a continuous-time node of the same type as in the last section.

Let us consider a variable B with state space $\mathscr{B} = \{b_1, \ldots, b_r\}$ which is preceded by a set of discrete variables $\mathbf{X}$ and one continuous-state variable A. Furthermore, let us assume we have full knowledge about probabilities of the type $p(B = b_i | A = t, \mathbf{X} = \mathbf{x})$ for all $b_i \in \mathscr{B}, t \in [0, \infty), \mathbf{x} \in \mathscr{X}$. Now it is possible, in principle, to extend the translation algorithm in a similar fashion as in the last subsection whenever we can compute quantities of the type $D(p(b|t,\mathbf{x}) \| p(b|t,\mathbf{x_s}))$ and $D(p(b|t,\mathbf{x}) \| p(b|\mathbf{x_s}))$.

In applications we sometimes want to make probabilistic statements about a number of possible 'consequence' random variables $\mathbf{B} = \{B_1, \ldots, B_r\}$ given information about a time-to-event variable A and other variables $\mathbf{X}$. For example, say the ignition time is bounded by some time t_b and the leak size was small, then certain fire types or other consequences are impossible or less likely to occur. These 'consequence' variables are often of binary nature and have a straight forward influence on each other or none at all.

While it is not impossible to handle dependencies of a B_i on $\mathbf{X}$, A and other B_j $(j \neq i)$, we restrict ourselves to the case where all of the B_i, $i = 1, \ldots, r$ only depend either on some subset of $\mathbf{X}$ and / or A. This can be a reasonable approximation in reality (compare also to Naive Bayes for classification settings, see e.g. (Hand and Yu, 2001)) and makes demonstration of the principle easier. Furthermore, we only consider the case $0 < A \leq t_{max}$ which corresponds to $\{A_0 = 1\}$, using the augmentation variable introduced earlier. This is because for the other cases distributions are either trivial or can be read off a table, such as was the case for the $\mathbf{X}$. For the instances of $\{A_0 = 0\}$ and $\{A_0 = 2\}$, the distributions of the $\{B_j\}_{j=1}^r$ are conditionally independent of A and can be directly specified using $\mathbf{X}$ and A_0. For convenience we will not explicitly express conditioning of B_j, $j = 1, \ldots, r$, on $\{0 < A \leq t_{max}\}$.

In DNV GL's risk model that we worked with, outputs of analyses contain discrete variables $\{B_j\}_{j=1}^r$ that are subsequent variables of the discretised A_d . For each variable B_j there exists a list of probabilities of the form $\{\mathbb{P}(B_j = b_j | A_d = I_i, \mathbf{X} = \mathbf{x})\}_{j=1}^{|\mathscr{B}_j|}$ for every $I_i \in \mathscr{A}_d$, $\mathbf{x} \in \mathscr{X}$.

After re-modelling A_d as a time-continuous version A, we are able to make statements about events conditioned on $\{t_\alpha < A \leq t_\beta\}$, where t_α, t_β need not be elements of the set T. In order to be coherent when introducing the $\mathbf{B}$, it is natural to require

$$\mathbb{P}(B_j = b_j, A_d = I_i | \mathbf{X} = \mathbf{x}) = \mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] | \mathbf{X} = \mathbf{x}) \tag{3.7}$$

for every $b_j \in \mathscr{B}_j$, $I_i \in \mathscr{A}_d$, $\mathbf{x} \in \mathscr{X}$ to hold.

By construction of A from A_d we automatically fulfil $\mathbb{P}(A_d = I_i | \mathbf{X} = \mathbf{x}) = \mathbb{P}(A \in (t_{i-1}, t_i] | \mathbf{X} = \mathbf{x})$

and $\mathbb{P}(A_d = I_i, \mathbf{X} = \mathbf{x}) = \mathbb{P}(A \in (t_{i-1}, t_i], \mathbf{X} = \mathbf{x})$. To fulfil Equation 3.7 we only require that

$$\mathbb{P}(B_j = b_j, A_d = I_i, \mathbf{X} = \mathbf{x}) = \mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i], \mathbf{X} = \mathbf{x}).$$

To achieve this requirement note that

$$\begin{aligned} \mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) &= \int_{t_{i-1}}^{t_i} p_{B_j, A|\mathbf{X}}(b_j, t|\mathbf{x})\, dt \\ &= \int_{t_{i-1}}^{t_i} p_{B_j, A|\mathbf{X}}(b_j|t, \mathbf{x})\, p_{A|\mathbf{X}}(t|\mathbf{x})\, dt, \end{aligned}$$

where $p_{A|\mathbf{X}}(t|\mathbf{x})$ was developed in the last section.

There are different choices to model $p_{B_j, A|\mathbf{X}}(b_j|t, \mathbf{x})$. One obvious choice would be to set

$$p_{B_j, A|\mathbf{X}}(b_j|t, \mathbf{x}) = p_{B_j, A_d|\mathbf{X}}(b_j|I_i, \mathbf{x}),$$

for $t \in I_i$ which just resembles the probabilities from the ET. To demonstrate the flexibility of allowing a more complicated form for our method, we choose a polynomial form

$$p_{B_j, A|\mathbf{X}}(b_j|t, \mathbf{x}) = c_i (t - t_{i-1})^{\rho - 1} + v_{i-1}, \tag{3.8}$$

for $t \in (t_{i-1}, t_i]$, where $c_i \in \mathbb{R}$ and v_{i-1} is short notation for $p_{B_j, A|\mathbf{X}}(b_j|t_{i-1}, \mathbf{x})$. The parameter ρ can be chosen freely.

Note the special cases of $\rho = 1$ and $\rho = 2$ which lead to $p_{B_j|A,\mathbf{X}}$ that are constant or linear on each time interval I_i, respectively. Using Equation 3.8 leads to

$$\mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) = \int_{t_{i-1}}^{t_i} \left(c_i (t - t_{i-1})^{\rho - 1} + v_{i-1} \right) p_{A|\mathbf{X}}(t|\mathbf{x})\, dt.$$

We determined the density $p_{A|\mathbf{X}}(t|\mathbf{x})$ before which was given by

$$p_{A|\mathbf{X}}(t|\mathbf{x}) = K h_{\mathbf{x}}(t)\, e^{-\int_0^t h_{\mathbf{x}}(\tilde{t}) d\tilde{t}}.$$

Hence

$$\mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) = \int_{t_{i-1}}^{t_i} \left(c_i (t - t_{i-1})^{\rho - 1} + v_{i-1} \right) K h_{\mathbf{x}}(t)\, e^{-\int_0^t h_{\mathbf{x}}(\tilde{t}) d\tilde{t}} dt.$$

The computation of the last integral can be found in Appendix (A.5) and results in

$$\begin{aligned} &\mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) = \\ &\qquad K \phi_{\mathbf{x}}(t_{i-1}) \left(c_i h_{\mathbf{x},i}^{1-\rho} \left[\Gamma(\rho) - \gamma(\rho, h_{\mathbf{x},i}(t_i - t_{i-1})) \right] + v_{i-1} \left(1 - e^{-h_{\mathbf{x},i}|I_i|} \right) \right). \end{aligned} \tag{3.9}$$

The above coherence condition (3.7) allows us to determine the parameter c_i (for fixed $b_j, \mathbf{x}$)

necessary to specify the probabilities $p_{B_j,A|\mathbf{X}}(b_j|t,\mathbf{x})$.

Let us recall that the probabilities of the type $\mathbb{P}(B_j = b_j|A_d = I_i, \mathbf{X} = \mathbf{x})$ are stored in the event tree as numbers that we might call $q_{b_j|I_i,\mathbf{x}}$. Now, to fulfil the coherence (3.7), we equate

$$\begin{aligned}
\mathbb{P}(B_j = b_j, A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) &= \mathbb{P}(B_j = b_j, A_d = I_i|\mathbf{X} = \mathbf{x}) \\
&= \mathbb{P}(B_j = b_j|A_d = I_i, \mathbf{X} = \mathbf{x})\,\mathbb{P}(A_d = I_i|\mathbf{X} = \mathbf{x}) \\
&= \mathbb{P}(B_j = b_j|A_d = I_i, \mathbf{X} = \mathbf{x})\,\mathbb{P}(A \in (t_{i-1}, t_i] \,|\mathbf{X} = \mathbf{x}) \\
&= q_{b_j|I_i,\mathbf{x}} \int_{t_{i-1}}^{t_i} p_{A|\mathbf{X}}(t|\mathbf{x})\, dt. \qquad (3.10)
\end{aligned}$$

Then for the event $A \in (t_i, t_{i+1}]$, using (3.9), Equation 3.10 translates to

$$\begin{aligned}
K\phi_{\mathbf{x}}(t_{i-1})\left(c_i h_{\mathbf{x},i}^{1-\rho}\left[\Gamma(\rho) - \gamma(\rho, h_{\mathbf{x},i}(t_i - t_{i-1}))\right] + v_{i-1}\left(1 - e^{-h_{\mathbf{x},i}|I_i|}\right)\right) &= \\
q_{b_j|I_i,\mathbf{x}} K\left(1 - e^{-h_{\mathbf{x},i},|I_i|}\right)\left(e^{-\sum_{j=1}^{i-1} h_j|I_j|}\right) &,
\end{aligned}$$

so that we obtain

$$c_i = \frac{\left(q_{b_j|I_i,\mathbf{x}} - v_0 - \sum_{j=1}^{i-1} c_j|I_j|^{\rho-1}\right)\left(1 - e^{-h_{\mathbf{x},i}|I_i|}\right)}{h_i^{1-\rho}\left[\Gamma(\rho) - \gamma(\rho, h_{\mathbf{x},i}|I_i|)\right]}.$$

Remark 3.10. *In the last equation we made use of the relation* $v_i = v_0 + \sum_{j=1}^{i} c_j|I_j|^{\rho-1}$.

After determination of the c_i, we have all ingredients to include the variables $\mathbf{B}$ into the translation network. It was mentioned earlier and is seen in the following Algorithm 5 that it is necessary to calculate quantities of the type $D(p(b|t,\mathbf{x}) \,\|\, p(b|t,\mathbf{x_s}))$ and $D(p(b|t,\mathbf{x}) \,\|\, p(b|\mathbf{x_s}))$. We have derived these expressions in Appendix (A.5) for the case $\rho = 2$ which we assume is fixed from now on.

Remark 3.11. *We remark here that the simplifications using the continuous-time extensions should be affected at different thresholds than simplifications using only discrete variables. We have built the new variable types by fulfilling some coherence conditions (more precisely Equation 3.5 and Equation 3.7), namely that certain conditional probabilities for the given time intervals are the same before and after the extension. This does not necessarily lead to the same magnitudes when calculating the KLD and CMI. Hence translations developed in this chapter should have a different sensitivity of edge removals to thresholds.*

In this chapter we set out to present two possibilities to extend time-discretised variables in ETs to time-continuous variables in a corresponding, translated BN. We introduced ignition time as an example that allows to incorporate a continuous variable into the translation framework by using purely analytical methods. We have found that children of such continuous nodes become hard to work with even though they might have a simple structure. This is not an uncommon problem in hybrid BNs. However, it is possible to include these nodes into the translation framework as well if one resorts to numerical methods. Further work on the intricacy of such methods would

Algorithm 5 Inclusion of a specific type of discrete variable with continuous underlying set into the network.

Prerequisite: The network translation for $\mathbf{X}$ and A has been carried according to Algorithms 2 and 4.

Input: Hazard functions $h_{\mathbf{x}}$ for all $\mathbf{x} \in \mathscr{X}$, $p(X_j|\mathrm{pa}(X_j))$ for $j = 1,\ldots,d$.

Output: Parent set $\mathrm{pa}(B)$, such that $CMI\,(B, \{(\mathbf{X} \cup A) \setminus \mathrm{pa}(B)\}|\mathrm{pa}(B)) \leq \alpha$.

1. Assign an initial set of potential parents $\mathrm{pa}_{\mathrm{pot}}(B) := \{A =: X_{d+1}, X_d, \ldots, X_1\}$.
2. Initialise a set of essential parents: $\mathrm{ess}\,(B) := \{\}$.
3. While $\left(\mathrm{pa}_{\mathrm{pot}}(B) \setminus \mathrm{ess}\,(B) \neq \emptyset\right)$ do:
   (a) For each $X_k \in \mathrm{pa}_{\mathrm{pot}}(B) \setminus \mathrm{ess}(B)$ do:
      i. Compute the necessary parameters as described in this section.
      ii. Compute $\alpha_k := CMI(B, \{\mathbf{X} \cup A\} \setminus \{\mathrm{pa}_{\mathrm{pot}}(B) \setminus X_k\}|\mathrm{pa}_{\mathrm{pot}}(B) \setminus X_k)$.
      iii. If $\alpha_k > \alpha : \mathrm{ess}(B) \leftarrow \mathrm{ess}(B) \cup X_k$.
   (b) If $(\{\alpha_k : \alpha_k \leq \alpha\} \neq \emptyset)$: Remove the $X_{\tilde{k}}$ for which $\tilde{k} = \mathrm{argmin}_k\{\alpha_k : \alpha_k \leq \alpha\}$ from $\mathrm{pa}_{\mathrm{pot}}(B)$:
   $$\mathrm{pa}_{\mathrm{pot}}(B) \leftarrow \mathrm{pa}_{\mathrm{pot}}(B) \setminus X_{\tilde{k}}.$$
   Else: Stop and go to step 4.
4. Set $\mathrm{pa}(B) \leftarrow \mathrm{ess}(B)$.

be required. The information of the extended models could be stored in tables that contain the calculated parameters instead of CPTs in discrete BNs. In many cases the number of these parameters should be equal to (hazard rates) or not much larger than the number of corresponding ET probabilities. The validity of our approach does not generally rely on special functional structures; there is some flexibility of using other transformations.

CHAPTER 4

SIMPLIFICATION OF EVENT TREES / BAYESIAN NETWORKS USING AN IMPACT-WEIGHTED INFORMATION MEASURE

In the last chapters we have translated and approximated graphical structures based on their probabilistic specifications. The guiding principle was that it is reasonable to simplify / approximate a conditional distribution whenever the information content in the approximation was similar, or, in other words, the information loss based on some entropy related measure was acceptable.

Our information measure of choice was the (expected) KLD which has been utilised in countless applications for its properties. However, the KLD also possesses a trait that could lead to difficulties when it is used for approximations which are controlled by bounding ratios of probabilities instead of differences. We present the following result from (Chan and Darwiche, 2005) which sheds more light on this.

Example 4.1. *Let P,Q be distributions on $\mathscr{X} = \{\omega_1, \omega_2, \omega_3\}$ in the following way: $P(\omega_1) = p$, $P(\omega_2) = q - p$, $P(\omega_3) = 1 - q$ and $Q(\omega_1) = kp$, $Q(\omega_2) = q - kp$, $Q(\omega_3) = 1 - q$, where p, q, k are constants such that $0 \leq p \leq q \leq 1$ and $0 \leq k \leq \frac{q}{p}$. Then*

$$D(P\|Q) = -p \log k - (q-p) \log \frac{q - kp}{q - p}.$$

For $p \to 0$, we get $D \to 0$.

The behaviour of D as in Example 4.1 is in itself not disadvantageous. If one is concerned with comparing probabilistic structures, especially in the sense of evaluating differences as a measure of approximation goodness, then this does not pose a problem. In safety risk applications often times the impact of an event is inversely related to the likelihood of it. Then the ratio of two probabilities would be more relevant.

We usually not only compare probabilistic structures but also associated impacts of events. If one is presented not only with a probabilistic structure but additionally with an outcome-associated

payoff or loss function, then this extra structure should be taken into account. Consider the following game as a thought experiment.

Example 4.2. *Suppose there are two coins called A and B. Coin A is fair such that 'heads' (H) and 'tails' (T) are equally likely outcomes, i.e.* $p(A = H) = p(A = T) = 0.5$. *Coin B is slightly biased in the sense that* $p(B = H) = 0.49$, $p(B = T) = 0.51$. *If we know nothing else, we might be happy to exchange one coin for the other as they should be similar in nature. Now suppose we are forced into a game, flipping coin A. It pays 1 unit if 'heads' comes up and costs 2 units if 'tails' comes up. Do we still want to exchange one coin for the other? Going to extremes, what if the coins were extremely biased but 'similar in probability', for example such that* $p(A = H) = 0.995, p(A = T) = 0.005$ *and* $p(B = H) = 0.99, p(B = T) = 0.01$, *but the payoff of the game is gaining 1 unit for 'heads' but losing 1000 units for 'tails'?*

So far we have measured the irrelevance of a potential parent variable X on a variable Y, given variables $\mathbf{Z}$ by comparing the information contents of $p(Y|\mathbf{Z},X)$ and $p(Y|\mathbf{Z})$ using the expected KLD. Hence, the decision to deem a variable irrelevant was based on the idea that the 'probabilistic structures are similar'. This is a reasonable approach, but for applications where there are losses or weights associated with outcomes, the situation might look different and then one could think back of a definition of risk as per (1.1) and imagine the idea of 'risk structures being similar'.

In this chapter we present an approach related to weighted entropy that allows to utilise loss information in our translation procedure. (This also has the potential to avoid the problem from Example 4.1; if we compare the likelihoods of events in the context of their impact, we would obtain a quantity that signifies if one of the events' impact trumps the other in a large scale.)

All variables in this chapter are discrete.

4.1 The weighted versions of information measures

In this section we introduce weighted versions and some of the properties of information theoretic quantities used earlier. It turns out that weighted information measures bring some additional complications; we also mention some possible modifications.

4.1.1 Simple extension of the classical information measures

A weighted version of the Shannon-entropy goes at least back to (Guiaşu, 1971) and can be defined as follows.

Definition 4.3. *(Weighted entropy (WE))*
Let X be a real-valued random variable with values in a finite set $\mathscr{X}$ and $w(x) : \mathscr{X} \to \mathbb{R}_0^+$ be a non-negative weight function. We define the weighted entropy $H_w(X)$ as

$$H_w(X) = -\sum_{x \in \mathscr{X}} w(x)p(x)\log p(x).$$

Weighted entropy was applied in different settings such as for judging investment risk (Nawrocki and Harding, 1986) or ranking the activity of chemicals (Shockley, 2014).

The above definition can be modified accordingly to obtain the joint weighted entropy and conditional weighted entropy. We borrow from (Suhov et al., 2016), but use versions for the discrete case.

Definition 4.4. *(Joint weighted entropy, conditional weighted entropy)*
Let X,Y be discrete random variables and w be a non-negative weight function, as before. The joint weighted entropy $H_w(X,Y)$ of X and Y is defined as

$$H_w(X,Y) = -\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}} w(x,y)p_{X,Y}(x,y)\log p_{X,Y}(x,y)$$

and the conditional weighted entropy $H_w(Y|X)$ of Y given X as

$$H_w(Y|X) = -\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}} w(x,y)p_{X,Y}(x,y)\log p_{Y|X}(y|x).$$

Closely related to weighted entropy are the weighted versions of the KLD and CMI.

Definition 4.5. *(Weighted KLD)*
Let X,Y be two discrete random variables with mass functions p,q on $\mathscr{X}$ respectively and let w be a non-negative weight function. We define the weighted Kullback-Leibler divergence $D_w(p\|q)$ between p and q by $w(x)$ as

$$D_w(p\|q) = \sum_{x\in\mathscr{X}} w(x)p(x)\log\frac{p(x)}{q(x)}.$$

Definition 4.6. *(Weighted CMI)*
Let X,Y,Z be discrete random variables with mass functions p_X, p_Y, p_Z on $\mathscr{X}$, $\mathscr{Y}$, $\mathscr{Z}$ respectively and let $w(x,y,z)$ be a non-negative weight function. We define the weighted CMI between X and Y given Z as

$$CMI_w(X,Y|Z) = \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w(x,y,z)p_{X,Y,Z}(x,y,z)\log\frac{p_{X,Y,Z}(x,y,z)\,p_Z(z)}{p_{X,Z}(x,z)\,p_{Y,Z}(y,z)}.$$

Remark 4.7. *We can also express the above as $CMI_w(X,Y|Z) = \mathbb{E}_{P_{Y,Z}} D_w(p(x|y,z)\,\|\,p(x|z))$.*

Similarly to Theorem 2.18, one can show that the weighted KLD can be used to establish a similarity between distributions, but one needs an extra assumption here. The next result, including its proof, is an adaptation for discrete variables from what can be found in (Suhov et al., 2016) again.

Theorem 4.8.

Let p, q and w be as above. We assume that

$$\sum_{x \in \mathscr{X}} w(x)\,[p(x) - q(x)] \geq 0. \tag{4.1}$$

Then

$$D_w(p\|q) \geq 0$$

and

$$D_w(p\|q) = 0 \Leftrightarrow \left(\frac{q(x)}{p(x)} - 1\right) w(x) = 0$$

for all $x \in \mathscr{X}$.

Proof. Let $S := \{x \in \mathscr{X} : p(x) > 0\}$. Then

$$\begin{aligned}
-D_w(p\|q) &= -\sum_{x \in S} w(x)p(x) \log \frac{p(x)}{q(x)} \\
&= \sum_{x \in S} w(x)p(x) \log \frac{q(x)}{p(x)} \\
&\leq \sum_{x \in S} w(x)p(x) \left[\frac{q(x)}{p(x)} - 1\right] \\
&= \sum_{x \in S} w(x)\,[q(x) - p(x)] \\
&\leq \sum_{x \in \mathscr{X}} w(x)\,[q(x) - p(x)] \\
&\leq 0.
\end{aligned}$$

□

Remark 4.9. *The theorem implies for our type of applications that if the weighted KLD is zero (or reasonably small), then for any $x \in \mathscr{X}$ either the associated losses $w(x)$ are zero (or negligible) or the mass functions for this point are exactly equal $p(x) = q(x)$ (or approximately $p(x) \approx q(x)$).*

Assumption 4.1 can be violated in such a way that $D_w(p\|q) < 0$. With distributions P, Q on $\{x_1, x_2\}$ with $P(x_1) = 0.51, P(x_2) = 0.49$ and $Q(x_1) = 0.5, Q(x_2) = 0.5$ and a weight function w with $w(x_1) = 0.1$ and $w(x_2) = 1$, one obtains $D_w(p\|q) < 0$. A negative D_w might not necessarily be impossible to deal with, however it seems unintuitive. In the case of comparing conditional distributions as we did before, it might mean we gain information by removing a variable. Still, small values of the weighted KLD, whether positive or negative, potentially could be interpreted as information content in the two distributions being 'close'. This still prevents us from trying to follow the same approach as in Section 2.3.2 to build more intuition about threshold selection. More research would be required to make sure what implications can be expected.

For less general weight functions $\tilde{w}$ it is possible to show that $D_{\tilde{w}}$ is non-negative. (Pocock, 2012) proved that the weighted mutual information for two variables is non-negative if the weight function only depends on one variable. We can make a similar assumption and present a special

case for which the weighted CMI is non-negative. The following lemma is an adaptation from (Pocock, 2012).

Lemma 4.10. *Let $w(x,y,z) = w(x)$ be a weight function that only depends on x.*
Then $CMI_w(X,Y|Z) \geq 0$.

Proof.

$$\begin{aligned} CMI_w(X,Y|Z) &= \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w(x,y,z)p(x,y,z)\log\frac{p(x,y,z)\,p(z)}{p(x,z)\,p(y,z)} \\ &= \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w(x)p(x,y,z)\log\frac{p(y|x,z)}{p(y|z)} \\ &= \sum_{x\in\mathscr{X}}\sum_{z\in\mathscr{Z}} w(x)p(x,z)\sum_{y\in\mathscr{Y}} p(y|x,z)\log\frac{p(y|x,z)}{p(y|z)} \\ &= \sum_{x\in\mathscr{X}}\sum_{z\in\mathscr{Z}} w(x)p(x,z)D(p(y|x,z)\|p(y|z)) \geq 0. \end{aligned}$$

□

In the last section of this chapter we will also use a weight function that fulfils such a restriction in order to do numerical comparisons.

4.1.2 Alternative ways to incorporate weights into information measures

We have seen in the previous subsection that general weight functions pose some problems for straightforward extensions of the KLD and MI.

(Kvålseth, 2017) suggests a modification $C\tilde{M}I(X,Y|Z)$ of conditional mutual information which leads to non-negativity and equality to zero if and only if the respective conditional independence is given. This modification leads also to non-negativity of a weighted version $C\tilde{M}I_w(X,Y|Z)$ and makes use of the basic inequality $-\log c + c - 1 > 0$ for $c > 0$. It looks like the following:

$$C\tilde{M}I_w(X,Y|Z) = CMI_w(X,Y|Z) + \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w(x,y,z)\left[p(z)\,p(x|z)\,p(y|z) - p(x,y,z)\right]. \quad (4.2)$$

A different way of incorporating weights for the comparison of two conditional distributions relies on rescaling the distributions by the weights, instead of introducing a definition of weighted CMI. Suppose we are given the joint mass function $p(x,y,z)$ and a weight function $w(x,y,z)$. We might define a scaled $\tilde{p}_w(x,y,z)$ as

$$\tilde{p}_w(x,y,z) = \frac{w(x,y,z)\,p(x,y,z)}{\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w(x,y,z)\,p(x,y,z)}$$

and use it as the basis of any relevant *CMI* computations. This $\tilde{p}_w(x,y,z)$ can be interpreted as a joint distribution that has been adjusted for the importance of outcomes by weights w. We might

infer any other needed quantities from $\tilde{p}_w(x,y,z)$. For example, $\tilde{p}_w(y,z) = \sum_{x\in\mathscr{X}} \tilde{p}_w(x,y,z)$. Then a w-scaled version $CMI_{s,w}$ of the CMI can be calculated as

$$CMI_{s,w}(X,Y|Z) = \sum_{x\in\mathscr{X}} \sum_{y\in\mathscr{Y}} \sum_{z\in\mathscr{Z}} \tilde{p}_w(x,y,z) \log \frac{\tilde{p}_w(x,y,z)\,\tilde{p}_w(z)}{\tilde{p}_w(x,z)\,\tilde{p}_w(y,z)}. \tag{4.3}$$

We will leave $\tilde{CMI}_w(X,Y|Z)$ for further research and consider the $CMI_{s,w}(X,Y|Z)$ as part of the numerical experiment in the last section of this chapter. Next we describe some different type of weight functions and considerations on modifying the translation algorithm.

4.2 Adjustments to the translation algorithm and weight function considerations

4.2.1 Two different types of weight functions

We can think of two types of possible weight functions. One type is given from external data, such as measured costs or damage estimates based on a physical model. These weights require a data source and probably do not have a specific functional structure. In order to check Condition 4.1 for each pair of mass functions that will be compared would require many tests. In addition, such data is usually only available for leaf nodes of a tree since those nodes represent the end of a sequence of outcomes that describes an accident scenario. We are doing sequential calculations across the tree starting near the root and therefore will need to have weights available not only for leaf nodes but for any generation of the tree. A simple and reasonable method to obtain this data is to work with the weights at the leafs and take expectations backwards through the tree. To be more precise, consider the following ET part in Figure 4-1 consisting of one branch only.

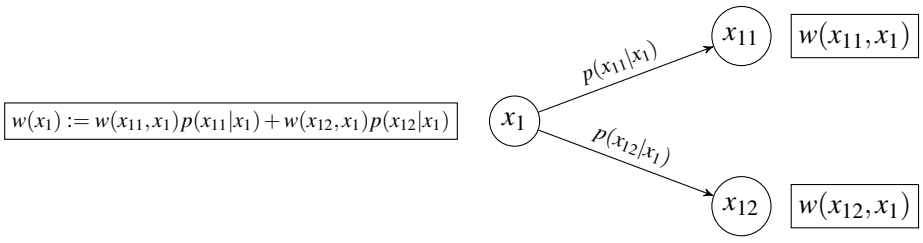

Figure 4-1: *Weighted loss aggregated as an expected value.*

We have losses $w(x_{11},x_1)$ and $w(x_{12},x_1)$ given for the full sequence of outcomes (x_1,x_{11}) and (x_1,x_{12}) respectively. Given only x_1, we know that we will incur loss $w(x_{11},x_1)$ with probability $p(x_{11}|x_1)$ and loss $w(x_{12},x_1)$ with probability $p(x_{12}|x_1)$. Thus, we decide to aggregate this information by taking the expected value and define

$$w(x_1) := w(x_{11},x_1)p(x_{11}|x_1) + w(x_{12},x_1)p(x_{12}|x_1).$$

By the same mechanism one can populate all generations of a tree with weights before doing any simplifications.

The other type of weight function is distribution dependent. (Guiaşu, 1971) names these weight functions 'objective' as they do not rely on anything else than the distribution itself. One of the examples used in the context of weighted entropy is $w(x) = \frac{-p(x)}{\log p(x)}$, where $\log p(x)$ is interpreted as the amount of information from a probability. We mentioned previously that in QRA the impact of an event is often related inversely to its probability. Following this train of thought one could choose a distribution dependent weight function $w_1(x,y,z) = \frac{1}{p(y,z)}$ which leads to a non-negative and simpler expression for $CMI_w(X,Y|Z)$:

$$\begin{aligned} CMI_{w_1}(X,Y|Z) &= \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} w_1(x,y,z)\,p(x,y,z)\log\frac{p(x,y,z)\,p(z)}{p(x,z)\,p(y,z)} \\ &= \sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} \frac{1}{p(y,z)}\,p(x,y,z)\log\frac{p(x,y,z)\,p(z)}{p(x,z)\,p(y,z)} \\ &= \sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} \frac{1}{p(y,z)}\,p(y,z)\sum_{x\in\mathscr{X}} p(x|y,z)\log\frac{p(x|y,z)}{p(x|z)} \\ &= \sum_{y\in\mathscr{Y}}\sum_{z\in\mathscr{Z}} D\left(p(x|y,z)\,\|\,p(x|z)\right). \end{aligned}$$

4.2.2 A weighted algorithm based on Assumption 4.1

The modification we now propose would essentially make Algorithm 3 of Chapter 2 a special case for a weight function $w \equiv 1$, however for other less trivial cases of w, we need to be cautious about Condition 4.1 again. Either it is required to check whether Condition 4.1 is fulfilled or one needs to choose a special type weight function w that automatically leads to non-negativity of CMI_w.

A possible way of using the weighted CMI to the purpose of local simplifications is presented in Algorithm 6. It is held in the language of Chapter 2.

In the next section some simple comparisons are performed based on the unweighted translation and the scaled translation using $CMI_{s,w}$.

4.3 Numerical experiments

In this section we carry out a numerical experiment using the event tree data set from Subsection 2.4.1. We choose a very simple weight function which depends only on `Conseq` and is described by $w(\texttt{Conseq} = 1) = 1$ and $w(\texttt{Conseq} = 2) = 2$. This is not an unreasonable assumption, it can be thought that harm or loss of life largely depends on a final outcome such as whether there is an explosion or which type of explosion. We rescale the joint mass function of the ET in Appendix A.2 as described in Equation 4.3. The resulting weight-scaled tree data can be seen in Appendix A.4. By comparison of Figure A-1 and Figure A-3 it is noticeable that the weighting seems to affect variables closer to the end of the tree more than variables closer to the root of the tree.

Algorithm 6 Determination of a simplified parent set for a variable Y.

Prerequisite: Condition 4.1 is fulfilled or the weight function $w(y,\mathbf{x})$ is chosen such that CMI_w is always non-negative.
Input: Distributions $p(Y|X_{i-1},\ldots,X_1)$, $p(X_j|\mathrm{pa}(X_j))$ for $j=1,\ldots,i-1$. Weight function $w(y,\mathbf{x})$.
Output: Parent set $\mathrm{pa}(Y)\subseteq\{X_1,X_2,\ldots,X_{i-1}\}$, such that
$CMI_w\left(Y,\{X_1,X_2,\ldots,X_{i-1}\}\setminus\mathrm{pa}(Y)\,|\mathrm{pa}(Y)\right)\leq\alpha$.

1. Assign an initial set of potential parents $\mathrm{pa}_{\mathrm{pot}}(Y):=\{X_{i-1},\ldots,X_1\}$.
2. Initialise a set of essential parents: $\mathrm{ess}(Y):=\{\}$.
3. While $\left(\mathrm{pa}_{\mathrm{pot}}(Y)\setminus\mathrm{ess}(Y)\neq\emptyset\right)$ do
   (a) For each $X_k\in\mathrm{pa}_{\mathrm{pot}}(Y)\setminus\mathrm{ess}(Y)$ do:
      i. Compute $\alpha_k:=CMI_w\left(Y,\{X_1,X_2,\ldots,X_{i-1}\}\setminus\left\{\mathrm{pa}_{\mathrm{pot}}(Y)\setminus X_k\right\}|\mathrm{pa}_{\mathrm{pot}}(Y)\setminus X_k\right)$
      ii. If $\alpha_k>\alpha$: $\mathrm{ess}(Y)\leftarrow\mathrm{ess}(Y)\cup X_k$.
   (b) If $(\{\alpha_k:\alpha_k\leq\alpha\}\neq\emptyset)$: Remove the $X_{\tilde{k}}$ for which $\tilde{k}=\mathrm{argmin}_k\{\alpha_k:\alpha_k\leq\alpha\}$ from $\mathrm{pa}_{\mathrm{pot}}(Y)$:
   $$\mathrm{pa}_{\mathrm{pot}}(Y)\leftarrow\mathrm{pa}_{\mathrm{pot}}(Y)\setminus X_{\tilde{k}}.$$
   Else: Stop and go to step 4.
4. Set $\mathrm{pa}(Y)\leftarrow\mathrm{ess}(Y)$.

Figure 4-2 shows as a first diagnostic the number of edges against different thresholds. Similarly to before α was varied in the interval $[0,0.001]$, such that $\alpha\in\{0+k\cdot0.00001\}$. (Due to numerical imprecisions, we actually start with $\alpha=10^{-10}$ instead of zero.)

The first thing we notice is that for the smallest choice of α, we have more edges using weight-scaled translation. The reason behind this could be that edges have been introduced where conditional independencies are present. Observe Figure 4-3 which displays the obtained networks for $\alpha\approx0$. The network resulting from using the weight-scaled mass function, on the right, contains a further link between `LeakSize` and `Weather` and between `SafetySys` and `Weather`. By construction of the toy example this now shows that additional parent relationships can be created besides the presence of a conditional independence. This is a particularity of using re-scaled mass functions in the translation. It might be reasonable for such a situation to simply ignore additional edges. We further add that these additional edges get removed quickly as the simplification threshold is raised to just $\alpha=0.00001$.

The weight-scaled translation seems to always lag the unweighted translation in terms of removing edges. We can compare this with the intuition of 'accounting for weights leads to stricter removal decisions'. Let us mention that the 'weakest link' is found between `SafetySys` and `IgnitionTime` for both methods. It could be possible to hypothesise that for not too wildly varying weight functions the order of edge removals does not change for both methods only the inclusion of weights makes edges more 'robust' to removals.

In this chapter we have motivated why the inclusion of available impact data might be rea-

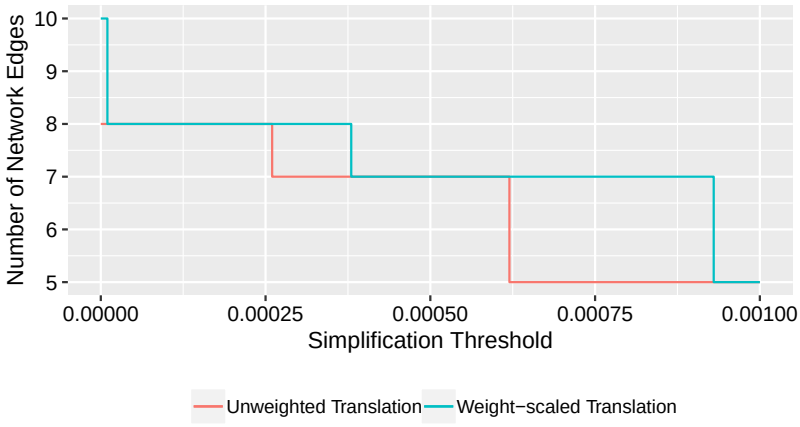

Figure 4-2: *Number of network edges for the toy example using unweighted and scaled translation.*

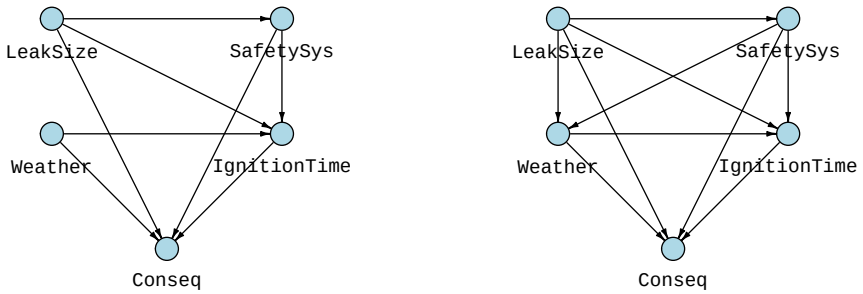

Figure 4-3: *Illustration of the obtained network structures for* $\alpha = 1 \cdot 10^{-10}$ *using unweighted translation (left) and using a weight-scaled translation (right).*

sonable before models are simplified based on probabilistic structures alone. The introduction of weighted information measures leads to some mathematical and computational intricacies. We presented some of the approaches that have been suggested in the literature; usually with certain restrictions. We have touched upon a few points in the example of this last section that could motivate to explore further the properties of weighted information measures such as the ones introduced. In relation to that, of interest are also conditions for weight functions that can be used consistently in a sequential manner through an ET.

CHAPTER 5

CONCLUDING REMARKS AND OUTLOOK

5.1 Summary

In this work we have investigated the connection between ETs and BNs by automatically translating the former into the latter. Motivated by the applications DNV GL presented, our main goal was to fit ETs used in specific areas of QRA into the more flexible framework of BNs, while at the same time allowing to simplify the probabilistic structures in certain ways. More specifically, the following has been done.

We haven given an introduction to the problem and presented the two main objects that we are concerned with, ETs and BNs, in Chapter 1.

In Chapter 2 we suggested the basic algorithm which allows the automatic translation of ETs to BNs and additionally extends previous algorithms by allowing to quantify strengths of dependencies and to simplify the resulting structure. The algorithm is sequential in the sense that it takes an order of the ET variables and creates a BN by adding these variables one by one according to the chosen order. Since the ETs we were working with could be seen as largely having a 'natural' order or representing a physical progression of events, we did restrict ourselves to the order that was given by the order of generations in the ET. In difference to previous publications we did not explicitly test for the equality of conditional distributions to find conditional independencies. Instead, we used tools from information theory (such as the Kullback-Leibler divergence and conditional mutual information) to quantify the dissimilarity between conditional distributions. This algorithm works with a threshold that determines acceptable information loss for each potential parent set. It has the advantage that many candidate parents can be detected immediately. We found a relation between the threshold in terms of information theoretic magnitude and the total variation; this can aid a practitioner to decide on a threshold value. We have tested this algorithm on a small-scale artifical ET and on data from a real-life case presented by DNV GL. We have found that it is important to be careful about numerical imprecisions but a simple two-step procedure was demonstrated to avoid such a problem. It was shown that this algorithm can equally be

used for simplifications on BNs directly; this was tested on three different BNs that have appeared in the literature earlier. It can be used to detect the strengths of dependencies between nodes and their parents, such as for finding the 'weakest' or 'strongest' link in a network.

In the next chapter we did consider two possible extensions for the translation of ETs to BNs. We focussed on re-modelling ignition time variables since they often play an important role in applications. In the encountered ETs we find information of a discretised version of an ignition time variable that comprises of ignition probabilities over certain fixed time intervals. Firstly, we used this information to create a continuous time variable based on piece-wise constant hazard functions. It was shown how this time-continuous variable can be embedded within the same translation method used in Chapter 2 using only discrete variables. One of the advantages of using a continuous-time variable over a discretised variable is the possibility of computing ignition probabilities over various, arbitrary intervals. Secondly, we outlined a way of including special type of child variables of ignition time into the translation algorithm. Typically the event of an ignition causes certain direct consequences, such as different types of explosions. We limited ourselves to the case where such consequence variables were extended into depending on a continuous underlying time set and probabilities for certain consequence types were of polynomial form. Even the investigation for these confined conditions showed the complexity of dealing mixed variables in BNs. The implementation for this setup needs careful numerical considerations.

In Chapter 4 we presented considerations about the inclusion of impact information for comparisons of the probabilistic structures. We introduced different versions of weighted information measures and discussed some properties for them. Finally, we compared results for translation without and with a special type of weighted information measure. An example data set illustrated the sensitivity of order of removal of parent relations as well as dependency strengths.

5.2 Outlook

There is a variety of potential further research questions and approaches to them.

The field of event trees has shown some efforts to incorporate uncertainty notions and soft computation methods instead of working with classical, crisp probabilities. For example, (You and Tonon, 2012) propose methods to work with imprecise probabilities for certain cases of information specifications within the ET and (Ferdous et al., 2009) explore fuzzy-set and evidence theory approaches to deal with data uncertainties in ETA. To draw a connection to the generalisation of the translation / simplification of ETs and BNs, it could be worth investigating ways of defining new information measures or modifying existing ones to account for uncertainty concepts. One could potentially work with imprecise probabilities and attempt to conceptualise an idea of 'imprecise conditional independence'. [1] Possible starting points could be (Bronevich and Klir, 2010), providing an investigation of axiomatic foundations for uncertainty measures for im-

[1] Connected to this is the question of how to properly compare sets of probability values with each other. One rough idea is that a random variable X could be regarded 'imprecise conditionally dependent' on Y, given Z, whenever a value set associated with X given Y and Z dominates in some sense the value set associated with X given Z.

precise probabilities, or (Lotfi and Fallahnejad, 2010) using a Shannon type entropy method for imprecise data, such as fuzzy sets.

The assumptions made for the model extension in Chapter 3 were motivated by some of the real-world example data and chosen in order to reduce the complexity of computations. Relaxing these assumptions, e.g. by introducing more of the continuous type variables and / or allowing more general dependencies between them will undoubtedly lead to vastly more complicated computations, both for model simplifications and potential inference queries. However, we can imagine that there are cases where strong interest only lies in predictive inference, such as the following: evidence is available and entered only for nodes without parents ('input nodes') and the only nodes for which the posterior distribution is of importance are nodes without children ('output nodes'). In this case it might be very possible to carry out exact inference besides the fact that the model structure has become more complicated. A possible examination of this problem could be inspired by further consulting with practitioners to determine the structure of reasonable special cases.

A different applications-related question concerns the weight function used in a weighted entropy approach. Since the weights represent the impact or importance of an outcome, they should be chosen in an appropriate way. There might be different interpretations of the so obtained information theoretic quantities that also determine the choice of a threshold. Additionally, the sensitivity of obtained quantities with respect to the choice of scaling factors and type of weight functions (for example, using empirically obtained 'utility values' or also weight functions depending on the probability distribution directly, as described in his original article (Guiaşu, 1971)) could be worthwhile to investigate, also to judge the robustness of the method.

One could consider a number of other algorithm-related questions. In this work we have always assumed a given joint probability distribution as our starting point. In a setting where distributions have to be learned or estimated from data, the simplification framework can be adjusted in such a way as to compare empirical conditional distributions. For a single step of determining a parent set for a variable, this can be compared to (Koller and Sahami, 1996) in which a method to select a subset of features for a classifier based on the expected KLD is described.

The common theme of the algorithms presented in this project was that of sequential translation / simplification methods. In Chapter 2 we touched upon the problem that errors introduced in a local approximation in a step i will have an impact on the decision for local approximations at a later step $j > i$. This means that approximation decisions made at some point will not be reversed, i.e. there is no feedback in later iterations that deems an earlier approximation unsuitable based on the overall picture. In contrast to the method used in (Minka, 2005), we wanted to find simplifications 'on the go' by controlling the effect of each single simplification; this differs from minimising some total error. It could provide useful to find a relation that expresses the connection between local errors and global errors in order to assess the overall approximation in a better way.

APPENDIX A

APPENDIX

A.1 An enumerative description of event trees

The intuition behind this construction is the following. Typically accidents start with an *initial event* C_0 which occurs with a certain frequency and triggers a sequence of other events $(C_1, \ldots, C_n)$ which might or might not be consequences of C_0. The word event can be somewhat misleading here, as the C_i are random variables in the mathematical sense. We will call these events / random variables consequences C_i $(i = 1, \ldots, n)$ and assume that each C_i has a variety of m_i different possible outcomes called $o_{i1}, o_{i2}, \ldots, o_{im_i}$ which are observed according to some probability distribution. (For now we will discuss the C_i in terms of its outcome set $\{o_{i1}, o_{i2}, \ldots, o_{im_i}\}$ and also use the language that "C_i consists of its possible outcomes".)

The case of a consequence consisting of only two possible outcomes can be interpreted as a Bernoulli random variable or a binary variable. This usually represents a consequence with a yes / no character, i.e. a particular event happens or not. (For example, the non- / occurrence of an explosion.) On the other hand, the case $m_i > 2$ usually represents a distinction of a degree or effect size of the consequence happening. (For example, no overpressure, small overpressure, large overpressure, etc.) We should note that $m_i = 1$ means that the outcome of the consequence is completely determined and hence the inclusion of consequence C_i into a model may be unnecessary.

If we count the initial event and all possible combinations of outcomes of the consequences, then such a model consists of $m_1 m_2 \ldots m_n$ many different sequences in total. Those sequences describe all possible courses of events in the analysis. Each of these sequences is completely described by a tuple of observed outcomes, e.g.

$$(C_1, C_2, C_3, \ldots, C_n) = (o_{13}, o_{21}, o_{31}, \ldots, o_{n5}).$$

We let the initial event C_0 correspond to the root node of the tree, which we denote by o_0. Then we create m_1 many nodes $\{o_{11}, o_{12}, \ldots, o_{1m_1}\}$ for the first generation of the tree. This

set of nodes will also be denoted o_1 and represents all the outcomes C_1 can take. For generation two, we will create a total of $m_1 m_2$ many nodes (the set of which will be denoted as o_2), which are formed by m_1 copies of the nodes $\{o_{21}, o_{22}, \ldots, o_{2m_2}\}$ and could be numbered as $\{\tilde{o}_{11}, \tilde{o}_{12}, \ldots, \tilde{o}_{1m_2}, \tilde{o}_{21}, \tilde{o}_{22}, \ldots, \tilde{o}_{2m_2}, \tilde{o}_{31}, \ldots, \tilde{o}_{m_1 m_2}\}$. Those represent all possible outcomes of C_2. We continue in this fashion until reaching generation n where we create $m_1 m_2 \ldots, m_n$ many nodes (the set of which also being denoted as o_n), consisting of $m_1 m_2 \ldots, m_{n-1}$ many copies of nodes $\{o_{n1}, o_{n2}, \ldots, o_{nm_n}\}$, numbered as

$$\{\tilde{o}_{111\ldots1}, \tilde{o}_{111\ldots2}, \ldots, \tilde{o}_{111\ldots m_n}, \tilde{o}_{121\ldots1}, \ldots, \tilde{o}_{121\ldots m_n}, \ldots, \tilde{o}_{m_1 11\ldots1}, \ldots, \tilde{o}_{m_1 m_2 m_3 \ldots m_n}\}$$

and corresponding to the possible outcomes of C_n. The edge set of this tree can be described in terms of the numbering we just introduced. We have an edge between all pairs of nodes of the type $\left(\tilde{o}_{(\mathbf{i}_s)}, \tilde{o}_{(\mathbf{i}_s,k)}\right)$ where $\mathbf{i}_s \in \times_{l=1}^{i}\{1,2,\ldots,m_l\}$ and $k \in \{1,2,\ldots,m_{i+1}\}$. In other words: the j-th node in generation i is connected by edges only to the nodes that make up the j-th copy of o_{i+1} in generation $i+1$.

Before we introduce probabilities onto this tree structure, we remark two things here. Firstly, using the notation o_{ij}, we do not have a unique enumeration of all nodes within the same generation, however every tuple $(o_0, o_{1j_1}, o_{2j_2}, \ldots, o_{nj_n})$ describes a unique path through the tree; from root to a leave node. That property allows us to continue using the o_{ij} notation without being too ambiguous. Secondly, the tree we have just described could be seen as a 'full' tree. It may be the case that occurrence of a certain outcome for C_i makes some of the outcomes for C_{i+1} impossible, e.g. observing $C_i = o_{ik}$ causes $C_{i+1} \in \underline{C} \subsetneq o_{i+1}$. Looking at this from a different perspective it means we may have introduced too many edges in the tree, some of which may display impossible connections. This is not a real problem since we will attach probability zero to such edges; this is explained in the following paragraph. (If one could ever know for certain whether or not any connections should be deemed impossible for a real-life model is another question.)

We add a probabilistic structure to the model by specifying a conditional probability for each edge of the tree. Those probabilities describe the likelihood of observing the event $\{C_{i+1} = o_{(i+1)j}\}$ given the observations for all preceding variables, that is, conditioned on $\{C_1 = o_{1j_1}, C_2 = o_{2j_2}, \ldots, C_i = o_{ij_i}\}$. Hence every edge joining some node in generation i and some node in generation $i+1$ is associated with the conditional probability

$$\mathbb{P}\left(C_{i+1} = o_{(i+1)j_{(i+1)}} \middle| \mathbf{C}_{[1:i]} = \mathbf{o}_{j_1 j_2 \cdots j_i}\right) := \mathbb{P}\left(C_{i+1} = o_{(i+1)j_{(i+1)}} \middle| C_1 = o_{1j_1}, C_2 = o_{2j_2}, \ldots C_i = o_{ij_i}\right).$$

By way of this construction it is possible to calculate the joint probability of a certain chain of consequences: Notice that for any discrete joint distribution for variables $C_1, C_2, \ldots, C_n$ it holds the general factorisation

$$\mathbb{P}(C_1, C_2, \ldots, C_n) = \prod_{i=1}^{n} \mathbb{P}(C_i | C_1, \ldots, C_{i-1}), \tag{A.1}$$

whenever this is defined. For the ET model, all the given edge probabilities resemble the factors in that last product of (A.1). Hence we can calculate every joint probability (the probability of a certain chain of consequence outcomes) by simply multiplying the conditional probabilities along the path representing that outcome chain:

$$\mathbb{P}\left(\mathbf{C}_{[1:n]} = \mathbf{o}_{j_1 j_2 \dots j_n}\right) = \prod_{i=0}^{n-1} \mathbb{P}\left(C_{i+1} = o_{(i+1)j_{(i+1)}} \,\middle|\, \mathbf{C}_{[1:i]} = \mathbf{o}_{j_1 j_2 \dots j_i}\right). \tag{A.2}$$

We see that we could assign probability zero to an edge which connects outcomes that are impossible to follow each other. Then all the relevant joint probabilities will be zero as well and so this process is similar as erasing any edge that represents consequence outcomes that are impossible to follow each other.

A.2 Artificial event tree table for Section 2.4.1

LeakSize	PLeakSize	SafetySys	PSafetySys	Weather	Pweather	IgnitionTime	PIgnitionTime	Conseq	Pconseq
1	0.8	1	0.95	1	0.52	0	0.01	1	0.72
1	0.8	1	0.95	1	0.52	0	0.01	2	0.28
1	0.8	1	0.95	1	0.52	1	0.011	1	0.78
1	0.8	1	0.95	1	0.52	1	0.011	2	0.22
1	0.8	1	0.95	1	0.52	2	0.01	1	0.8
1	0.8	1	0.95	1	0.52	2	0.01	2	0.2
1	0.8	1	0.95	1	0.52	3	0.969	1	1
1	0.8	1	0.95	1	0.52	3	0.969	2	0
1	0.8	1	0.95	0	0.48	0	0.014	1	0.7
1	0.8	1	0.95	0	0.48	0	0.014	2	0.3
1	0.8	1	0.95	0	0.48	1	0.021	1	0.67
1	0.8	1	0.95	0	0.48	1	0.021	2	0.33
1	0.8	1	0.95	0	0.48	2	0.023	1	0.5
1	0.8	1	0.95	0	0.48	2	0.023	2	0.5
1	0.8	1	0.95	0	0.48	3	0.942	1	1
1	0.8	1	0.95	0	0.48	3	0.942	2	0
1	0.8	0	0.05	1	0.52	0	0.013	1	0.55
1	0.8	0	0.05	1	0.52	0	0.013	2	0.45
1	0.8	0	0.05	1	0.52	1	0.022	1	0.67
1	0.8	0	0.05	1	0.52	1	0.022	2	0.33
1	0.8	0	0.05	1	0.52	2	0.027	1	0.95
1	0.8	0	0.05	1	0.52	2	0.027	2	0.05
1	0.8	0	0.05	1	0.52	3	0.938	1	1
1	0.8	0	0.05	1	0.52	3	0.938	2	0
1	0.8	0	0.05	0	0.48	0	0.025	1	0.51
1	0.8	0	0.05	0	0.48	0	0.025	2	0.49
1	0.8	0	0.05	0	0.48	1	0.031	1	0.49
1	0.8	0	0.05	0	0.48	1	0.031	2	0.51
1	0.8	0	0.05	0	0.48	2	0.033	1	0.68
1	0.8	0	0.05	0	0.48	2	0.033	2	0.32
1	0.8	0	0.05	0	0.48	3	0.911	1	1
1	0.8	0	0.05	0	0.48	3	0.911	2	0
2	0.2	1	0.9	1	0.52	0	0.015	1	0.6
2	0.2	1	0.9	1	0.52	0	0.015	2	0.4
2	0.2	1	0.9	1	0.52	1	0.02	1	0.62
2	0.2	1	0.9	1	0.52	1	0.02	2	0.38
2	0.2	1	0.9	1	0.52	2	0.019	1	0.69
2	0.2	1	0.9	1	0.52	2	0.019	2	0.31
2	0.2	1	0.9	1	0.52	3	0.946	1	1
2	0.2	1	0.9	1	0.52	3	0.946	2	0
2	0.2	1	0.9	0	0.48	0	0.021	1	0.54
2	0.2	1	0.9	0	0.48	0	0.021	2	0.46
2	0.2	1	0.9	0	0.48	1	0.029	1	0.52
2	0.2	1	0.9	0	0.48	1	0.029	2	0.48
2	0.2	1	0.9	0	0.48	2	0.028	1	0.5
2	0.2	1	0.9	0	0.48	2	0.028	2	0.5
2	0.2	1	0.9	0	0.48	3	0.922	1	1
2	0.2	1	0.9	0	0.48	3	0.922	2	0
2	0.2	0	0.1	1	0.52	0	0.02	1	0.55
2	0.2	0	0.1	1	0.52	0	0.02	2	0.45
2	0.2	0	0.1	1	0.52	1	0.03	1	0.55
2	0.2	0	0.1	1	0.52	1	0.03	2	0.45
2	0.2	0	0.1	1	0.52	2	0.033	1	0.6
2	0.2	0	0.1	1	0.52	2	0.033	2	0.4
2	0.2	0	0.1	1	0.52	3	0.917	1	1
2	0.2	0	0.1	1	0.52	3	0.917	2	0
2	0.2	0	0.1	0	0.48	0	0.031	1	0.41
2	0.2	0	0.1	0	0.48	0	0.031	2	0.59
2	0.2	0	0.1	0	0.48	1	0.042	1	0.35
2	0.2	0	0.1	0	0.48	1	0.042	2	0.65
2	0.2	0	0.1	0	0.48	2	0.05	1	0.3
2	0.2	0	0.1	0	0.48	2	0.05	2	0.7
2	0.2	0	0.1	0	0.48	3	0.877	1	1
2	0.2	0	0.1	0	0.48	3	0.877	2	0

Figure A-1: *Full event tree table for the artificial data set in Section 2.4.1.*

A.3 Event tree table for Example 2.35 in Section 2.4.1

A	pA	B	pB	C	pC	D	pD
1	0.105	1	0.918	1	0.138	1	0.708
1	0.105	1	0.918	1	0.138	2	0.292
1	0.105	1	0.918	2	0.862	1	0.088
1	0.105	1	0.918	2	0.862	2	0.912
1	0.105	2	0.082	1	0.238	1	0.084
1	0.105	2	0.082	1	0.238	2	0.916
1	0.105	2	0.082	2	0.762	1	0.224
1	0.105	2	0.082	2	0.762	2	0.776
2	0.895	1	0.899	1	0.482	1	0.093
2	0.895	1	0.899	1	0.482	2	0.907
2	0.895	1	0.899	2	0.518	1	0.862
2	0.895	1	0.899	2	0.518	2	0.138
2	0.895	2	0.101	1	0.53	1	0.624
2	0.895	2	0.101	1	0.53	2	0.376
2	0.895	2	0.101	2	0.47	1	0.527
2	0.895	2	0.101	2	0.47	2	0.473

Figure A-2: *Full event tree table for Example 2.35 in Section 2.4.1.*

A.4 Weight-scaled event tree table for Section 4.3

LeakSize	PLeakSize	SafetySys	PSafetySys	Weather	Pweather	IgnitionTime	PIgnitionTime	Conseq	Pconseq
1	0.797503172	1	0.94946473	1	0.516209075	0	0.012708246	1	0.5625
1	0.797503172	1	0.94946473	1	0.516209075	0	0.012708246	2	0.4375
1	0.797503172	1	0.94946473	1	0.516209075	1	0.013323802	1	0.639344262
1	0.797503172	1	0.94946473	1	0.516209075	1	0.013323802	2	0.360655738
1	0.797503172	1	0.94946473	1	0.516209075	2	0.011913981	1	0.666666667
1	0.797503172	1	0.94946473	1	0.516209075	2	0.011913981	2	0.333333333
1	0.797503172	1	0.94946473	1	0.516209075	3	0.96205397	1	1
1	0.797503172	1	0.94946473	1	0.516209075	3	0.96205397	2	0
1	0.797503172	1	0.94946473	0	0.483790925	0	0.017797248	1	0.538461538
1	0.797503172	1	0.94946473	0	0.483790925	0	0.017797248	2	0.461538462
1	0.797503172	1	0.94946473	0	0.483790925	1	0.027311931	1	0.503759398
1	0.797503172	1	0.94946473	0	0.483790925	1	0.027311931	2	0.496240602
1	0.797503172	1	0.94946473	0	0.483790925	2	0.033736542	1	0.333333333
1	0.797503172	1	0.94946473	0	0.483790925	2	0.033736542	2	0.666666667
1	0.797503172	1	0.94946473	0	0.483790925	3	0.921154279	1	1
1	0.797503172	1	0.94946473	0	0.483790925	3	0.921154279	2	0
1	0.797503172	0	0.05053527	1	0.514122805	0	0.018581314	1	0.379310345
1	0.797503172	0	0.05053527	1	0.514122805	0	0.018581314	2	0.620689655
1	0.797503172	0	0.05053527	1	0.514122805	1	0.028842931	1	0.503759398
1	0.797503172	0	0.05053527	1	0.514122805	1	0.028842931	2	0.496240602
1	0.797503172	0	0.05053527	1	0.514122805	2	0.027945902	1	0.904761905
1	0.797503172	0	0.05053527	1	0.514122805	2	0.027945902	2	0.095238095
1	0.797503172	0	0.05053527	1	0.514122805	3	0.924629852	1	1
1	0.797503172	0	0.05053527	1	0.514122805	3	0.924629852	2	0
1	0.797503172	0	0.05053527	0	0.485877195	0	0.035864898	1	0.342281879
1	0.797503172	0	0.05053527	0	0.485877195	0	0.035864898	2	0.657718121
1	0.797503172	0	0.05053527	0	0.485877195	1	0.045069419	1	0.324503311
1	0.797503172	0	0.05053527	0	0.485877195	1	0.045069419	2	0.675496689
1	0.797503172	0	0.05053527	0	0.485877195	2	0.041940267	1	0.515151515
1	0.797503172	0	0.05053527	0	0.485877195	2	0.041940267	2	0.484848485
1	0.797503172	0	0.05053527	0	0.485877195	3	0.877125416	1	1
1	0.797503172	0	0.05053527	0	0.485877195	3	0.877125416	2	0
2	0.202496828	1	0.897462216	1	0.51560846	0	0.020598535	1	0.428571429
2	0.202496828	1	0.897462216	1	0.51560846	0	0.020598535	2	0.571428571
2	0.202496828	1	0.897462216	1	0.51560846	1	0.02707236	1	0.449275362
2	0.202496828	1	0.897462216	1	0.51560846	1	0.02707236	2	0.550724638
2	0.202496828	1	0.897462216	1	0.51560846	2	0.024414168	1	0.526717557
2	0.202496828	1	0.897462216	1	0.51560846	2	0.024414168	2	0.473282443
2	0.202496828	1	0.897462216	1	0.51560846	3	0.927914938	1	1
2	0.202496828	1	0.897462216	1	0.51560846	3	0.927914938	2	0
2	0.202496828	1	0.897462216	0	0.48439154	0	0.029549529	1	0.369863014
2	0.202496828	1	0.897462216	0	0.48439154	0	0.029549529	2	0.630136986
2	0.202496828	1	0.897462216	0	0.48439154	1	0.041365485	1	0.351351351
2	0.202496828	1	0.897462216	0	0.48439154	1	0.041365485	2	0.648648649
2	0.202496828	1	0.897462216	0	0.48439154	2	0.040478806	1	0.333333333
2	0.202496828	1	0.897462216	0	0.48439154	2	0.040478806	2	0.666666667
2	0.202496828	1	0.897462216	0	0.48439154	3	0.88860618	1	1
2	0.202496828	1	0.897462216	0	0.48439154	3	0.88860618	2	0
2	0.202496828	0	0.102537784	1	0.509402153	0	0.028000386	1	0.379310345
2	0.202496828	0	0.102537784	1	0.509402153	0	0.028000386	2	0.620689655
2	0.202496828	0	0.102537784	1	0.509402153	1	0.042000579	1	0.379310345
2	0.202496828	0	0.102537784	1	0.509402153	1	0.042000579	2	0.620689655
2	0.202496828	0	0.102537784	1	0.509402153	2	0.044607512	1	0.428571429
2	0.202496828	0	0.102537784	1	0.509402153	2	0.044607512	2	0.571428571
2	0.202496828	0	0.102537784	1	0.509402153	3	0.885391523	1	1
2	0.202496828	0	0.102537784	1	0.509402153	3	0.885391523	2	0
2	0.202496828	0	0.102537784	0	0.490597847	0	0.04561397	1	0.257861635
2	0.202496828	0	0.102537784	0	0.490597847	0	0.04561397	2	0.742138365
2	0.202496828	0	0.102537784	0	0.490597847	1	0.064131632	1	0.212121212
2	0.202496828	0	0.102537784	0	0.490597847	1	0.064131632	2	0.787878788
2	0.202496828	0	0.102537784	0	0.490597847	2	0.078660732	1	0.176470588
2	0.202496828	0	0.102537784	0	0.490597847	2	0.078660732	2	0.823529412
2	0.202496828	0	0.102537784	0	0.490597847	3	0.811593666	1	1
2	0.202496828	0	0.102537784	0	0.490597847	3	0.811593666	2	0

Figure A-3: *Weight-scaled event tree table for the artificial data set in Section 4.3.*

A.5 Detailed calculations for the model extension in Chapter 3

For a fixed time point set $T = \{t_0, t_1, \ldots, t_m\}$ and a piece-wise constant function $h_{\mathbf{x}}(t)$ on $(0, t_m]$ with break points given from T and value $h_{\mathbf{x},i}$ on interval I_i, we define the following recurring expression:

$$\phi_{\mathbf{x}}(t) := e^{-\sum_{i=1}^{k} h_{\mathbf{x},i}|I_i| - h_{\mathbf{x},k+1}(t-t_k)},$$

where $|I_i| = |t_i - t_{i-1}|$ and $t_k = \max\{t_j; t_j \leq t\}$.

(A.5.1)

$$\int_0^t h_{\mathbf{x}}(\tilde{t})\, d\tilde{t} = \int_0^t \sum_{i=0}^{m-1} h_{\mathbf{x},i+1} \mathbb{I}\{t_i < \tilde{t} \leq t_{i+1}\}\, d\tilde{t} = \sum_{i=1}^{k} |I_i| h_{\mathbf{x},i} + (t - t_k) h_{\mathbf{x},k+1}.$$

(A.5.2)

$$\begin{aligned} D(f_{\mathbf{x}_1} \| f_{\mathbf{x}_2}) &= \int_0^{t_m} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt \\ &= \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt. \end{aligned}$$

Now we can evaluate each integral of the form $\int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt$.

Let $K_1 = \frac{1}{1-\phi_{\mathbf{x}_1}(t_m)}$ and $K_2 = \frac{1}{1-\phi_{\mathbf{x}_2}(t_m)}$ similar, then

$$\begin{aligned} \int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt &= \int_{t_i}^{t_{i+1}} K_1 h_{\mathbf{x}_1,i+1} \phi_{\mathbf{x}_1}(t) \log\left[\frac{K_1 h_{\mathbf{x}_1,i+1} \phi_{\mathbf{x}_1}(t)}{K_2 h_{\mathbf{x}_2,i+1} \phi_{\mathbf{x}_2}(t)}\right] dt \\ &= K_1 h_{\mathbf{x}_1,i+1} \phi_{\mathbf{x}_1}(t_i) \int_{t_i}^{t_{i+1}} e^{-h_{\mathbf{x}_1,i+1}(t-t_i)} \left(\log\left[\frac{K_1}{K_2}\right] + \log\left[\frac{h_{\mathbf{x}_1,i+1}}{h_{\mathbf{x}_2,i+1}}\right] \right. \\ &\quad \left. + \log\left[e^{\sum_{j=1}^{i}\left(h_{\mathbf{x}_2,j} - h_{\mathbf{x}_1,j}\right)|I_j|}\right] + \log\left[e^{\left(h_{\mathbf{x}_2,i+1} - h_{\mathbf{x}_1,i+1}\right)(t-t_i)}\right] \right) dt. \end{aligned}$$

If we let $C_{1,i} = K_1 \left(\log\frac{K_1}{K_2} + \log\frac{h_{\mathbf{x}_1,i+1}}{h_{\mathbf{x}_2,i+1}} + \sum_{j=1}^{i} |I_j| \left(h_{\mathbf{x}_2,j} - h_{\mathbf{x}_1,j}\right) \right)$ and $C_{2,i} = \frac{K_1\left(h_{\mathbf{x}_2,i+1} - h_{\mathbf{x}_1,i+1}\right)}{h_{\mathbf{x}_1,i+1}}$, then

$$\begin{aligned} \int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt &= C_{1,i}\left[\phi_{\mathbf{x}_1}(t_i) - \phi_{\mathbf{x}_1}(t_{i+1})\right] + C_{2,i}\left[\phi_{\mathbf{x}_1}(t_i) - \left(1 + h_{\mathbf{x}_1,i+1}|I_{i+1}|\right)\phi_{\mathbf{x}_1}(t_{i+1})\right] \\ &= \left(C_{1,i} + C_{2,i}\right)\phi_{\mathbf{x}_1}(t_i) - \left(C_{1,i} + C_{2,i}\left(\left(1 + h_{\mathbf{x}_1,i+1}|I_{i+1}|\right)\right)\right)\phi_{\mathbf{x}_1}(t_{i+1}). \end{aligned}$$

Summarising,

$$\begin{aligned} D\left(f_{\mathbf{x}_1} \| f_{\mathbf{x}_2}\right) &= \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} f_{\mathbf{x}_1}(t) \log\left[\frac{f_{\mathbf{x}_1}(t)}{f_{\mathbf{x}_2}(t)}\right] dt \\ &= \sum_{i=0}^{m-1} C_{1,i}\left[\phi_{\mathbf{x}_1}(t_i) - \phi_{\mathbf{x}_1}(t_{i+1})\right] + C_{2,i}\left[\phi_{\mathbf{x}_1}(t_i) - \left(1 + h_{\mathbf{x}_1,i+1}|I_{i+1}|\right)\phi_{\mathbf{x}_1}(t_{i+1})\right]. \end{aligned}$$

(A.5.3)

$$\begin{aligned} \int_{t_{i-1}}^{t_i} \left(c_i (t-t_{i-1})^{\rho-1} + v_{i-1}\right) K h_{\mathbf{x}}(t)\, e^{-\int_0^t h_{\mathbf{x}}(\tilde{t})d\tilde{t}} dt &= \\ K c_i h_{\mathbf{x},i} \int_{t_{i-1}}^{t_i} (t-t_{i-1})^{\rho-1} e^{-\int_0^t h_{\mathbf{x}}(\tilde{t})d\tilde{t}} dt + K h_{\mathbf{x},i} v_{i-1} \int_{t_{i-1}}^{t_i} e^{-\int_0^t h_{\mathbf{x}}(\tilde{t})d\tilde{t}} dt &= \\ K c_i h_{\mathbf{x},i} e^{-\sum_{j=1}^{i-1}|I_j|h_{\mathbf{x},j}} \int_{t_{i-1}}^{t_i} (t-t_{i-1})^{\rho-1} e^{-(t-t_{i-1})h_{\mathbf{x},i}} dt + K h_{\mathbf{x},i} v_{i-1} e^{-\sum_{j=1}^{i-1}|I_j|h_j} \int_{t_{i-1}}^{t_i} e^{-(t-t_{i-1})h_{\mathbf{x},i}} dt &= \\ K c_i h_{\mathbf{x},i} e^{-\sum_{j=1}^{i-1}|I_j|h_{\mathbf{x},j}} \int_{t_{i-1}}^{t_i} (t-t_{i-1})^{\rho-1} e^{-(t-t_i)h_{\mathbf{x},i}} dt + K v_{i-1} e^{-\sum_{j=1}^{i-1}|I_j|h_{\mathbf{x},j}} \left(1 - e^{-h_{\mathbf{x},i}|I_i|}\right) \end{aligned}$$

Let us evaluate the integral $\int_{t_{i-1}}^{t_i} (t-t_{i-1})^{\rho-1} e^{-(t-t_i)h_{\mathbf{x},i}} dt$ separately:

$$\begin{aligned} \int_{t_{i-1}}^{t_i} (t-t_{i-1})^{\rho-1} e^{-(t-t_i)h_{\mathbf{x},i}} dt &= \int_0^{|I_i|} z^{\rho-1} e^{-z h_{\mathbf{x},i}} dz \\ &= \frac{1}{h_{\mathbf{x},i}^{\rho}} \left[\Gamma(\rho) - \gamma\left(\rho, h_{\mathbf{x},i}(t_i - t_{i-1})\right)\right], \end{aligned}$$

with $\Gamma(\rho) = \int_0^\infty t^{\rho-1} e^{-t} dt$ being the gamma function and $\gamma(\rho, z) = \int_z^\infty t^{\rho-1} e^{-t} dt$ being the incomplete gamma function.

Combining the above results, we obtain

$$\begin{aligned} \int_{t_{i-1}}^{t_i} \left(c_i (t-t_{i-1})^{\rho-1} + v_{i-1}\right) K h_{\mathbf{x}}(t)\, e^{-\int_0^t h_{\mathbf{x}}(\tilde{t})d\tilde{t}} dt &= \\ K e^{-\sum_{j=1}^{i-1}|I_j|h_{\mathbf{x},j}} \left(c_i h_{\mathbf{x},i}^{1-\rho} \left[\Gamma(\rho) - \gamma\left(\rho, h_{\mathbf{x},i}(t_i - t_{i-1})\right)\right] + v_{i-1}\left(1 - e^{-h_{\mathbf{x},i}|I_i|}\right)\right). \end{aligned}$$

(A.5.4)

$$
\begin{aligned}
D\left(p\left(b|t,\mathbf{x}\right)\|p\left(b|t,\mathbf{x_s}\right)\right) &= \sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\int_0^{t_m} p\left(t|\mathbf{x}\right)\sum_{b\in\mathscr{B}} p\left(b|t,\mathbf{x}\right)\log\frac{p\left(b|t,\mathbf{x}\right)}{p\left(b|t,\mathbf{x_s}\right)}dt \\
&= \sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\int_0^{t_m} p\left(t|\mathbf{x}\right) \\
&\quad \sum_{b\in\mathscr{B}} \left(c_{i,1}\left(t-t_i\right)+v_{i-1,1}\right)\log\frac{c_{i,1}\left(t-t_i\right)+v_{i-1,1}}{c_{i,2}\left(t-t_i\right)+v_{i-1,2}}dt \\
&= \sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\sum_{b\in\mathscr{B}}\sum_{i=1}^{m} \\
&\quad \int_{t_{i-1}}^{t_i} p\left(t|\mathbf{x}\right)\left(c_{i,1}\left(t-t_i\right)+v_{i-1,1}\right)\log\frac{c_{i,1}\left(t-t_i\right)+v_{i-1,1}}{c_{i,2}\left(t-t_i\right)+v_{i-1,2}}dt \\
&= \frac{1}{1-e^{-\sum_{i=1}^{m} h_i|I_i|}}\sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\sum_{b\in\mathscr{B}}\sum_{i=1}^{m} e^{-\sum_{k=1}^{i-1} h_k|I_k|} \\
&\quad \int_{t_{i-1}}^{t_i} h_i e^{-h_i\left(t-t_{i-1}\right)}\left(c_{i,1}\left(t-t_i\right)+v_{i-1,1}\right)\log\frac{c_{i,1}\left(t-t_i\right)+v_{i-1,1}}{c_{i,2}\left(t-t_i\right)+v_{i-1,2}}dt.
\end{aligned}
$$

At this point we suspect that the last integral becomes analytically intractable and we would decide to solve it using a quadrature method.

$$
\begin{aligned}
D\left(p\left(b|t,\mathbf{x}\right)\|p\left(b|\mathbf{x}\right)\right) &= \sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\int_0^{t_m} p\left(t|\mathbf{x}\right)\sum_{b\in\mathscr{B}} p\left(b|t,\mathbf{x}\right)\log\frac{p\left(b|t,\mathbf{x}\right)}{p\left(b|\mathbf{x}\right)}dt \\
&= \sum_{\mathbf{x}\in\mathscr{X}} p\left(\mathbf{x}\right)\int_0^{t_m} p\left(t|\mathbf{x}\right) \\
&\quad \sum_{b\in\mathscr{B}} \left(c_{i,1}\left(t-t_i\right)+v_{i-1,1}\right)\log\frac{\left(c_{i,1}\left(t-t_i\right)+v_{i-1,1}\right)}{p\left(b|\mathbf{x}\right)}dt.
\end{aligned}
$$

This is a similar case where we also think that no analytical solution might be available and we would suggest using a simpler, numerical method.

BIBLIOGRAPHY

M. Abimbola, F. Khan, and N. Khakzad. Dynamic safety risk analysis of offshore drilling. *Journal of Loss Prevention in the Process Industries*, 30:74–85, 2014.

S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

L. M. Barclay, J. L. Hutton, and J. Q. Smith. Refining a Bayesian network using a chain event graph. *International Journal of Approximate Reasoning*, 54(9):1300–1309, 2013.

L. M. Barclay, R. A. Collazo, J. Q. Smith, P. A. Thwaites, and A. E. Nicholson. The dynamic chain event graph. *Electronic Journal of Statistics*, 9(2):2130–2169, 2015.

T. Bedford and R. Cooke. *Probabilistic risk analysis : Foundations and methods.* Cambridge University Press, Cambridge, 2001.

A. Bobbioa, L. Portinalea, M. Minichino, and E. Ciancamerla. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. *Reliability Engineering and System Safety*, 71(3):249–260, 2001.

A. Bronevich and G. J. Klir. Measures of uncertainty for imprecise probabilities: An axiomatic approach. *International Journal of Approximate Reasoning*, 51(4):365–390, 2010.

H. Chan and A. Darwiche. A distance measure for bounding probabilistic belief change. *International Journal of Approximate Reasoning*, 38(2):149–174, 2005.

A. Choi, H. Chan, and A. Darwiche. On Bayesian network approximation by edge deletion. *arXiv preprint arXiv:1207.1370 [cs.AI]*, 2012.

A. C. Constantinou, N. Fenton, and M. Neil. Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks. *Knowledge-Based Systems*, 50:60–86, 2013.

G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

V. T. Covello and J. Mumpower. Risk analysis and risk management: An historical perspective. *Risk Analysis*, 5(2):103–120, 1985.

T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley, Hoboken, 2nd edition, 2006.

I. Csiszár. Axiomatic characterizations of information measures. *Entropy*, 10(3):261–273, 2008.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

V. De Vasconcelos, W. A. Soares, and A. C. L. Da Costa. FN-curves: Preliminary estimation of severe accident risks after Fukushima. In *Proceedings of the Seventh International Nuclear Atlantic Conference (INAC 2015)*, volume 4, pages 2503–2514, 2015.

R. Diestel. *Graph theory.* Graduate Texts in Mathematics. Springer, Berlin, Heidelberg, 5th edition, 2017.

N. Fenton and M. Neil. *Risk assessment and decision analysis with Bayesian networks.* CRC Press, Boca Raton, 2013.

R. Ferdous, F. Khan, R. Sadiq, P. Amyotte, and B. Veitch. Handling data uncertainties in event tree analysis. *Process Safety and Environmental Protection*, 87(5):283–292, 2009.

P. Forré and J. M. Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv preprint arXiv:1710.08775 [math.ST]*, 2017.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29 (2-3):131–163, 1997.

G. L. Gilardoni. On Pinsker's and Vajda's type inequalities for Csiszár's f-divergences. *IEEE Transactions on Information Theory*, 56(11):5377–5386, 2010.

S. Guiaşu. Weighted entropy. *Reports on Mathematical Physics*, 2(3):165–179, 1971.

D. J. Hand and K. Yu. Idiot's Bayes: Not so stupid after all? *International Statistical Review*, 69 (3):385–398, 2001.

A. Hanea and D. Kurowicka. Mixed non-parametric continuous and discrete Bayesian belief nets. In T. Bedford, J. Quigley, L. Walls, B. Alkali, A. Daneshkhah, and G. Hardman, editors, *Advances in Mathematical Modeling for Reliability*, pages 9–16. IOS Press, Amsterdam, 2008.

D. Heckerman. A tutorial on learning with Bayesian networks. In D. E. Holmes and L. C. Jain, editors, *Innovations in Bayesian Networks : Theory and Applications*, pages 33–82. Springer, Berlin, Heidelberg, 2008.

D. Heckerman, E. J. Horvitz, and B. N. Nathwani. Toward normative expert systems: Part I. The Pathfinder project. *Methods of Information in Medicine*, 31(2):90–105, 1992.

S. Højsgaard. Graphical independence networks with the grain package for R. *Journal of Statistical Software*, 46(10):1–26, 2012.

E.-S. Hong, I.-M. Lee, H.-S. Shin, S.-W. Nam, and J.-S. Kong. Quantitative risk evaluation based on event tree analysis technique: Application to the design of shield TBM. *Tunnelling and Underground Space Technology*, 24(3):269–277, 2009.

T. Hrycej. Gibbs sampling in Bayesian networks. *Artificial Intelligence*, 46(3):351–363, 1990.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

N. Khakzad, F. Khan, and P. Amyotte. Dynamic safety analysis of process systems by mapping bow-tie into Bayesian network. *Process Safety and Environmental Protection*, 91(1-2):46–53, 2013.

U. Kjærulff. Reduction of computational complexity in Bayesian networks through removal of weak dependencies. In R. L. De Mantaras and D. Poole, editors, *Uncertainty Proceedings 1994*, pages 374–382. Morgan Kaufmann, San Francisco, 1994.

J. P. Klein and M. L. Moeschberger. *Survival analysis : Techniques for censored and truncated data*. Statistics for Biology and Health. Springer, New York, 2nd edition, 2003.

T. A. Kletz. *Learning from accidents*. Gulf Professional, Oxford, 3rd edition, 2001.

D. Koller and M. Sahami. Toward optimal feature selection. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 284–292. Morgan Kaufmann, San Francisco, 1996.

K. B. Korb and A. E. Nicholson. *Bayesian artificial intelligence*. CRC Press, Boca Raton, 2nd edition, 2011.

T. Koski and J. M. Noble. *Bayesian networks : An introduction*. Wiley, Oxford, 2009.

T. O. Kvålseth. On normalized mutual information: Measure derivations and properties. *Entropy*, 19(11):631, 2017.

H. Langseth, T. D. Nielsen, R. Rumí, and A. Salmerón. Inference in hybrid Bayesian networks. *Reliability Engineering and System Safety*, 94(10):1499–1509, 2009.

S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–194, 1988.

S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H. G. Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990.

J. F. Lawless. *Statistical models and methods for lifetime data*. Wiley, Hoboken, 2nd edition, 2003.

F. H. Lotfi and R. Fallahnejad. Imprecise Shannon's entropy and multi attribute decision making. *Entropy*, 12(1):53–62, 2010.

D. Margaritis. *Learning Bayesian Network Model Structure from Data*. Thesis (Ph.D.), Carnegie Mellon University, Pittsburgh, 2003.

D. W. R. Marsh and G. Bearfield. Generalizing event trees using Bayesian networks. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 222(2): 105–114, 2008.

V. Mihajlovic and M. Petkovic. Dynamic Bayesian networks: A state of the art. Technical Report TR-CTIT-34, University of Twente, Netherlands, 2001.

T. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, UK, 2005.

S. Moral, R. Rumí, and A. Salmerón. Mixtures of truncated exponentials in hybrid Bayesian networks. In S. Benferhat and P. Besnard, editors, *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 2001)*, Lecture Notes in Computer Science, vol 2143, pages 156–167. Springer, Berlin, Heidelberg, 2001.

D. N. Nawrocki and W. H. Harding. State-value weighted entropy as a measure of investment risk. *Applied Economics*, 18(4):411–419, 1986.

A. Neri, W. P. Aspinall, R. Cioni, A. Bertagnini, P. J. Baxter, G. Zuccaro, D. Andronico, S. Barsotti, P. D. Cole, T. Esposti Ongaro, T. K. Hincks, G. Macedonio, P. Papale, M. Rosi, R. Santacroce, and G. Woo. Developing an event tree for probabilistic hazard and risk assessment at Vesuvius. *Journal of Volcanology and Geothermal Research*, 178(3):397–415, 2008.

U. Nodelman, C. R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 378–387. Morgan Kaufmann, San Francisco, 2002.

Nuclear Regulatory Commission. Reactor safety study. An assessment of accident risks in US commercial nuclear power plants. Executive summary: Main report. Technical Report WASH-1400-MR, Nuclear Regulatory Commission, United States, 1975.

A. Oniśko. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*. Thesis (Ph.D.), Polish Academy of Science, Warsaw, 2003.

I. A. Papazoglou. Mathematical foundations of event trees. *Reliability Engineering and System Safety*, 61(3):169–183, 1998.

J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Francisco, 1988.

A. C. Pocock. *Feature Selection via Joint Likelihood*. Thesis (Ph.D.), University of Manchester, Manchester, 2012.

O. Pourret, P. Naïm, and B. Marcot, editors. *Bayesian networks: A practical guide to applications*. Wiley, 2008.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

M. Rausand. *Risk assessment: Theory, methods, and applications*. Statistics in practice. Wiley, Hoboken, 2011.

A. Rényi. On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, 1961.

T. Rosqvist, R. Molarius, H. Virta, and A. Perrels. Event tree analysis for flood protection - An exploratory study in Finland. *Reliability Engineering and System Safety*, 112:1–7, 2013.

A. Salmerón, R. Rumí, H. Langseth, T. D. Nielsen, and A. L. Madsen. A review of inference algorithms for hybrid Bayesian networks. *Journal of Artificial Intelligence Research*, 62:799–828, 2018.

I. Sason and S. Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62 (11):5973–6006, 2016.

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

M. Scutari and J.-B. Denis. *Bayesian networks: With examples in R*. Texts in Statistical Science. CRC Press, Boca Raton, 2015.

M. B. Sesen, A. E. Nicholson, R. Banares-Alcantara, T. Kadir, and M. Brady. Bayesian networks for clinical decision support in lung cancer care. *PLoS ONE [Online]*, 8(12):e82349, 2013.

G. Shafer. *Probabilistic expert systems*. Society for Industrial and Applied Mathematics, 1996.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, 1948.

P. P. Shenoy and J. C. West. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657, 2011.

K. R. Shockley. Using weighted entropy to rank chemicals in quantitative high-throughput screening experiments. *Journal of Biomolecular Screening*, 19(3):344–353, 2014.

Y. Suhov, I. Stuhl, S. Yasaei Sekeh, and M. Kelbert. Basic inequalities for weighted entropies. *Aequationes Mathematicae*, 90(4):817–848, 2016.

J. Thornton. Approximate inference of Bayesian networks through edge deletion. Thesis (M.Sc.), Kansas State University, Manhattan, 2005.

S. Tong and D. Koller. Active learning for parameter estimation in Bayesian networks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 647–653. MIT Press, 2001.

A. B. Tsybakov. *Introduction to nonparametric estimation*. Springer Series in Statistics. Springer, New York, 2009.

G. Unnikrishnan, Shrihari, and N. A. Siddiqui. Application of Bayesian methods to event trees with case studies. *Reliability: Theory & Applications*, 9(3):32–45, 2014.

R. A. Van Engelen. Approximating Bayesian belief networks by arc removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):916–920, 1997.

J. E. Vinnem. *Offshore risk assessment vol 2.: Principles, modelling and applications of QRA studies*. Springer Series in Reliability Engineering. Springer, London, 3rd edition, 2014.

W. D. Wallis. *A beginner's guide to graph theory*. Birkhäuser, Boston, 2nd edition, 2007.

X. You and F. Tonon. Event tree analysis with imprecise probabilities. *Risk Analysis*, 32(2): 330–344, 2012.